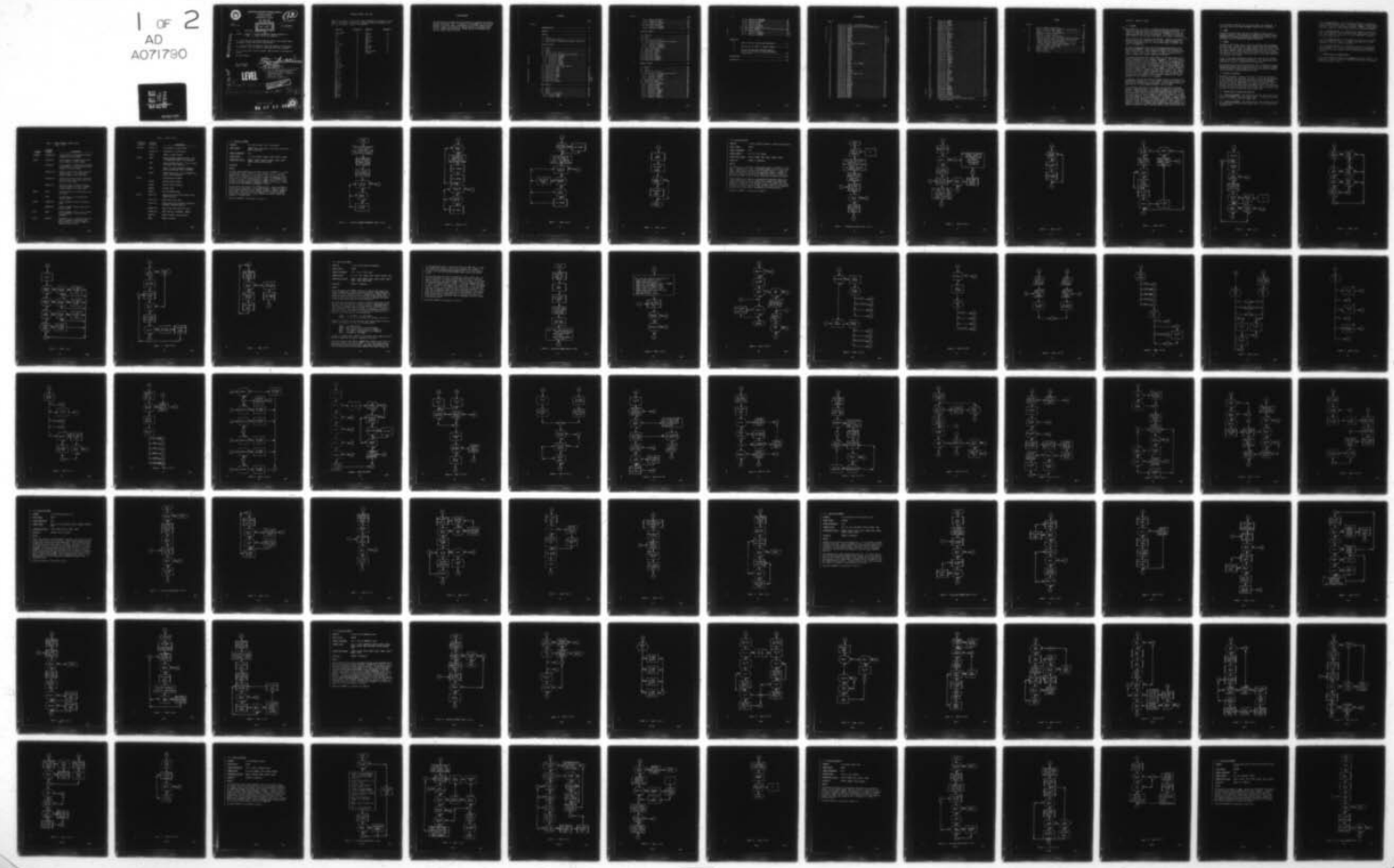
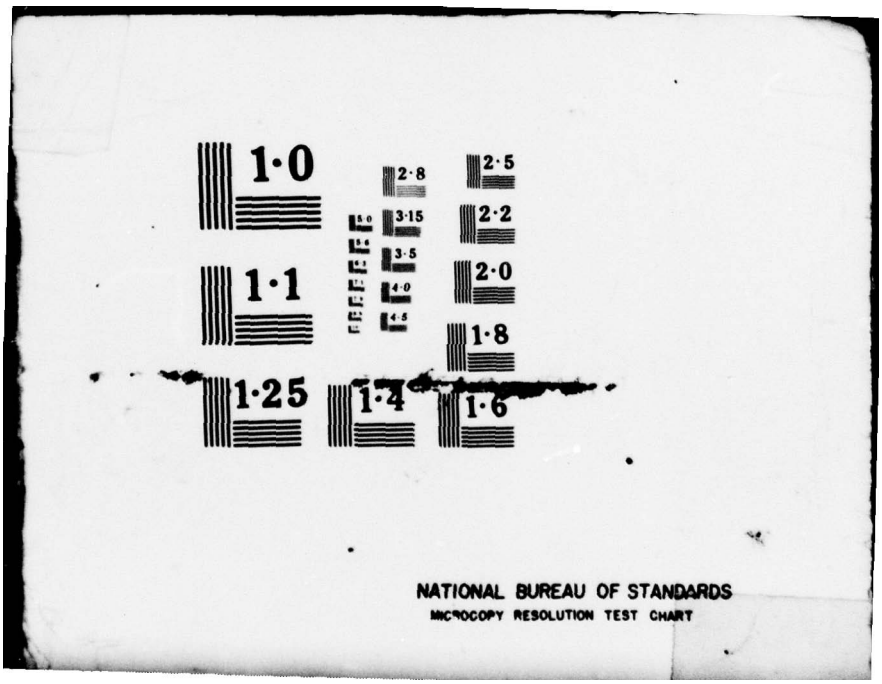
AD-A071 790 COMMAND AND CONTROL TECHNICAL CENTER WASHINGTON DC F/G 15/7
THE CCTC QUICK-REACTING GENERAL WAR GAMING SYSTEM (QUICK) PROGR--ETC(U)
JUN 79

UNCLASSIFIED CCTC-CSM-MM-99-77-VOL-3-C NL

1 OF 2
AD
A071790




1·0
1·1
1·25
2·8
3·15
3·5
4·0
4·5
2·5
2·2
2·0
1·8
1·4
1·6
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

DEFENSE COMMUNICATIONS AGENCY
COMMAND AND CONTROL
TECHNICAL CENTER
WASHINGTON, D. C. 20301

IN REPLY
REFER TO: C314

29 June 1979

TO: RECIPIENTS

SUBJECT: Change 1 to Program Maintenance Manual CSM MM 9-77, Volume III, Weapon Allocation Subsystem

1. Insert the enclosed change pages and destroy the replaced pages according to applicable security regulations.

2. A list of Effective Pages to verify the accuracy of this manual is enclosed. This list should be inserted before the title page.

3. When this change has been posted, make an entry in the Record of Changes.

FOR THE DIRECTOR:

166 Enclosures
Change 1 pages

J. DOUGLAS POTTER
Assistant to the Director
for Administration

The CCTC Quick-Reacting General War Gaming System (QUICK) Program Maintenance Manual. Volume III. Weapon Allocation Subsystem. Change 1.

EFFECTIVE PAGES - MAY 1979

This list is used to verify the accuracy of CSM MM 9-77, Volume III after change 1 pages have been inserted. Original pages are indicated by the letter 0 and change 1 pages by the numeral 1.

Page No.	Change No.	Page No.	Change No.
Title Page	0	371	1
ii-iii	1	372-374	0
iv	0	375	1
v-ix	1	376-398	0
x	0	399	1
1-5	0	400-409	0
6-60	1	410	1
60.1-60.34	1	411	0
61	0	412-413	1
62-64	1	414-421	0
65-85	0	422	1
86-87	1	422.1-422.10	1
88-89	0	423-516	0
90	1		
91-103	0		
104	1		
105-108	0		
109	1		
110-122	0		
123-126	1		
126.1-126.2	1		
127-154	0		
155	1		
156-165	0		
166	1		
167-170	0		
171	1		
172-176	0		
177	1		
178-179	0		
180	1		
181-183	0		
184-185	1		
186-197	0		
198	1		
199-201	0		
202	1		
203-364	0		
365-369	1		
370	0		

ACKNOWLEDGMENT

This document was prepared under the direction of the Chief for Military Studies and Analyses, CCTC, in response to a requirement of the Studies, Analysis, and Gaming Agency, Organization of the Joint Chiefs of Staff. Technical support was provided by System Sciences, Incorporated under Contract Number DCA100-75-C-0019. Change set one was prepared under Contract Number DCA100-78-C-0035.

CONTENTS

Section

ILLUSTRATIONS

TABLES

SECTION 2. PREPALOC MODULE

2.1 Purpose

The purpose of this module is to perform preliminary calculations on the weapon and target data as stored within the integrated data base. The output data from PREPALOC will be in a form convenient for use by the remaining processors of the plan generator. In addition, the user may select options to modify some of the data at this stage of processing.

Module PREPALOC has three major capabilities: updating of geographic and weapon group data, modification of target values and damage constraints and preparation of data for the fixed weapon assignment capability of program ALOC.

The basic raw geographic data must be data base defined prior to any execution of PREPALOC. Using this data, PREPALOC will calculate and store distances and attrition between all doglegs for use within processors to follow. Based on user inputs, the number of weapons within bomber or missile MIRV weapon groups may be adjusted.

The second major capability of this module is the modification of the target characteristics, VTO, MINKILL, and MAXKILL. VTO is the value of the target relative to all the others. MINKILL is the minimum fraction of target value that must be destroyed, and MAXKILL is the maximum desired fraction of target value destroyed. Any of these parameters may be changed for any target. The change requests can change these parameters for a single target or for a set of targets. The set of targets for which a change is requested is identified by target class, type, an individual identifier (target designator code (DESIG)) or any combination of these. For complex targets, the class, type, designator code, and index of each component will be checked to determine if a target parameter for the complex is to be changed.

In addition, the user can specify the height of burst to be used in any weapon/target combination. The user selects either a ground burst or an air burst at the optimal air burst height. In the absence of any user specification, the most damaging height of burst is used.

The third major capability is the request for allocation of specific weapons to specific targets. This fixing of weapons to targets enables the user to determine part of the weapon allocation while leaving the allocation module free to determine the remaining allocation. In addition, the time of arrival at target or launch salvo number can be fixed for missile weapons. This information will be passed to module PLANOUT which will adjust launch time accordingly. The fixing of weapons remains in effect for the remainder of the plan generation process. Later programs will retain the assignments as best possible. (For example, it is possible to fix a set of weapons from a weapon group with multiple independently targetable reentry vehicles (MIRV) in such a manner that there

are no feasible footprints that cover that target set adequately. In that case, some of the fixed assignment requests must be ignored.)

2.2 Input

The entire integrated data base must be completely defined prior to PREPALOC execution. This includes the storage of all targets, related geographic data, weapon type and group characteristics and other supporting data such as warhead and payload information.

2.3 Output

Creation of new records occurs if the user specified fixed assignments. For these cases, records called 'ASSIGN' which stores the fixed assignment under the proper target and weapon group linkage are created. Also, new records (RDDIST, TPDIST and TDDIST) defining the distance between each depenetration corridor recovery base intersection, each penetration corridor target intersection, and each target and the optimal depenetration corridor, are created.

If any of the target modification options are employed, the necessary target records will be modified accordingly. Also, weapon group attributes may be altered if overallocation is specified.

For all executions, distances and attrition rates associated with each penetration corridor will be calculated and stored. Similarly, depenetration distances between doglegs as well as the distance from depenetration corridor to recovery bases are stored.

2.4 Concept of Operation

PREPALOC is designed to operate in two modes: RECALC and non-RECALC. In the RECALC mode all geographic and weapon calculations are performed; in the non-RECALC mode these time consuming functions are bypassed. The user will ideally run PREPALOC once in the RECALC mode and then as many times in the non-RECALC mode as is required to make the target parameter changes and fixed weapon assignments needed. The print option functions are also separated such that optional prints may appear in any run once the data to be displayed has been developed.

2.5 Identification of Subroutine Functions

2.5.1 Subroutine GEOPREP. This subroutine performs the calculation and storage of geographic data which links targets to corridors and recovery bases. It is called only in the RECALC mode.

2.5.2 Subroutine WEPPREP. This subroutine calculates weapon group overallocation and sets up the salvoed weapon arrays. It is called only in the RECALC mode.

2.5.3 Subroutine SETRD. This subroutine is called in any execution which contains a SETTING clause. It performs changes to general gaming parameters and weapon group height of bursts. Changes to target parameters are stored on an external data file (25) for subroutine CHGBAS.

2.5.4 Subroutine CHGBAS. This subroutine is called in any execution where subroutine SETRD has stored one or more change requests on file 25. These requests are carried out for the target base according to a preset priority scheme.

2.5.5 Subroutine FIXWEP. This subroutine is called in any execution with one or more FIX clauses. It creates fixed assignment records (ASSIGN) for use by the allocation module.

2.5.6 Subroutine PRNPRP. This subroutine performs all standard and optional prints. It is called at the end of every run and produces those optional prints requested and any standard prints called for by the input.

2.6 PREPALOC Internal Common Blocks

All common blocks used internally by PREPALOC are given in table 1. For definition of common blocks that communicate with the COP, see Program Maintenance Manual, Volume I.

Table 1. Module PREPALOC Common Blocks
(Part 1 of 2)

COMMON	VARIABLE OR ARRAY	DESCRIPTION
CRLENGTH	CRLENGTH(30)	Total length of a penetration corridor; indexed by corridor number
DISTEF	DISTEF(50)	Total length of a depenetration corridor; indexed by corridor number
	DISTEG(50)	Total length of a depenetration corridor plus distance to nearest recovery base; indexed by corridor number
	DRECOV(50,4)	DESIG of recovery base associated with depenetration corridor; indexed by corridor number, up to four bases
	DISTNA(50,4)	Distance from depenetration corridor to recovery base; indexed by corridor number, up to four bases
	INDXA(50,4)	Ordinal of bases ordered on distance from depenetration corridor; indexed by corridor number, up to four bases
GEOREF	IPNRF	IDS reference code of penetration corridor header
	IDPRF	IDS reference code of depenetration corridor header
IFXREQ	IFXREQ(250)	Fixed assignment requests; indexed by group
	IFXHON(250)	Fixed assignment requests honored; indexed by group
IFXSW	FIXSW	Fixed assignment switch: true if there are assignments in this run, false otherwise
NTBFLG	NTBFLG(9)	Alphanumeric value to indicate source of values in general gaming parameters. May be DEFAUL for default, UNCHNG for charged during this run

Table 1. (Part 2 of 2)

ASSOCIATED COMMON	VARIABLE OR ARRAY	DESCRIPTION
RFPOINTS	RFLAT(20)	Latitutude of a refuel point
	RFLONG(20)	Longitude of a refuel point
	MREGN	Number of refuel points
SETCOM	NTBCH	General gaming parameter switch: true if changes were made, false otherwise
	DBCH	Target parameter switch: true if changes were made, false otherwise
	NDBCH	Number of target parameter change requests written onto file 25 by SETRD
	RECALC	RECALC mode switch: true if RECALC mode active, false otherwise
TSTUFF		Communicates with TOFM
	XTOFMIN	Minimum time of flight
	XCMISS	Missile flight parameter
	XRNGMIN	Minimum range
	XRANGE	Missile maximum range
WEPCOM	IWRCD(250)	Weapon group record IDS reference code; indexed on group
	IWTYP(250)	Weapon group type index
	IWCLS(250)	Weapon group class indicator (1=missile, 2=bomber, 3=salvoed missile)
	NMWTYP(120)	Weapon type name; indexed on type
	IWMIRV(250)	MIRV indicator (0=nonMIRV, 1=MIRV)
	NASM(250)	Number of ASMs in group payload
	NMGRP	Number of groups

2.7 Subroutine ENTMOD

PURPOSE: To control overall flow of processing

ENTRY POINTS: ENTMOD (first subroutine called when overlay link PREP is executed)

FORMAL PARAMETERS: None

COMMON BLOCKS: C15, C30, ERRCOM, IFXREQ, IFXSW, NTBFLG, SETCOM

SUBROUTINES CALLED: CHGBAS, FIXWEP, GEOPREP, HDFND, INSGET, MODFY, PRNPRP, RETRV, SETRD, WEPPREP

CALLED BY: COP

Method:

The basic method employed by the driver routine of PREPALOC is to check for particular input clauses in a set order and carry out functions based on their presence. As a preliminary step, the IDS error code "R04" is set as acceptable so that the retrieval of nonexistent DESIG values will not cause termination and the number table (NUMTBL) is retrieved. The input is then scanned for a RECALC clause, if it exists the general gaming parameters (INITSTRK, CORMSL, etc.) are set to their default values and GEOPREP and WEPPREP are called.

The subroutine now looks for any SETTING clauses. Subroutine SETRD is called for each occurrence. If any SETTING clause called for changes to the target parameters (VALUE, MINKILL, MAXKILL or IDHOB), CHGBAS is also called. Then, the input is scanned for FIX clauses and FIXWEP called at each occurrence. Finally, PRNPRP is called to produce any standard or optional print.

Subroutine ENTMOD is illustrated in figure 3.

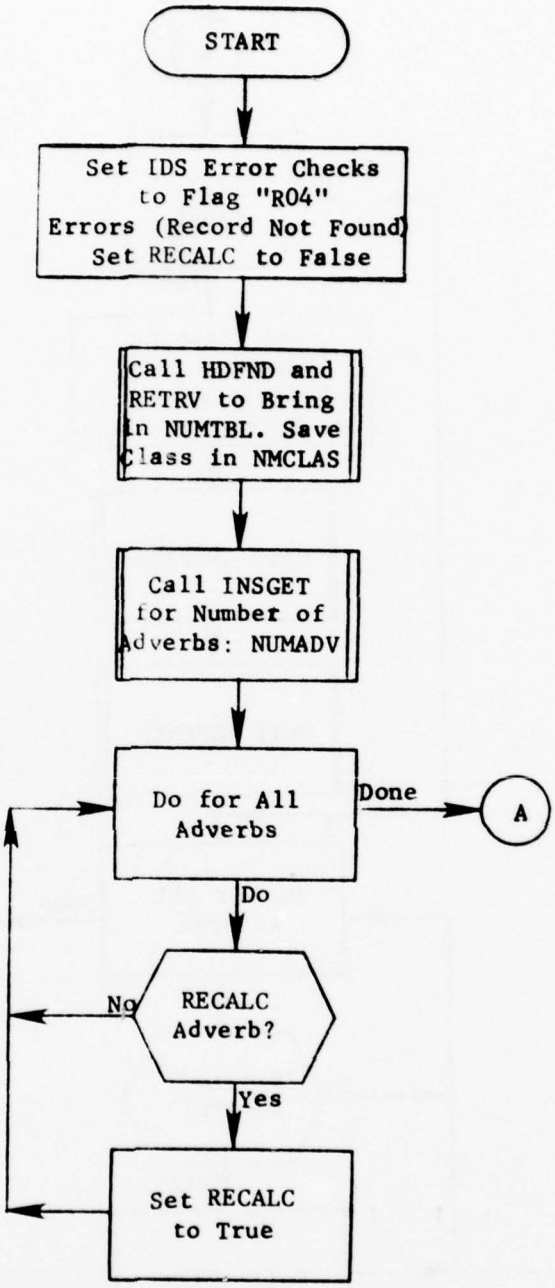


Figure 3. Subroutine ENTMOD (PREPALOC) (Part 1 of 4)

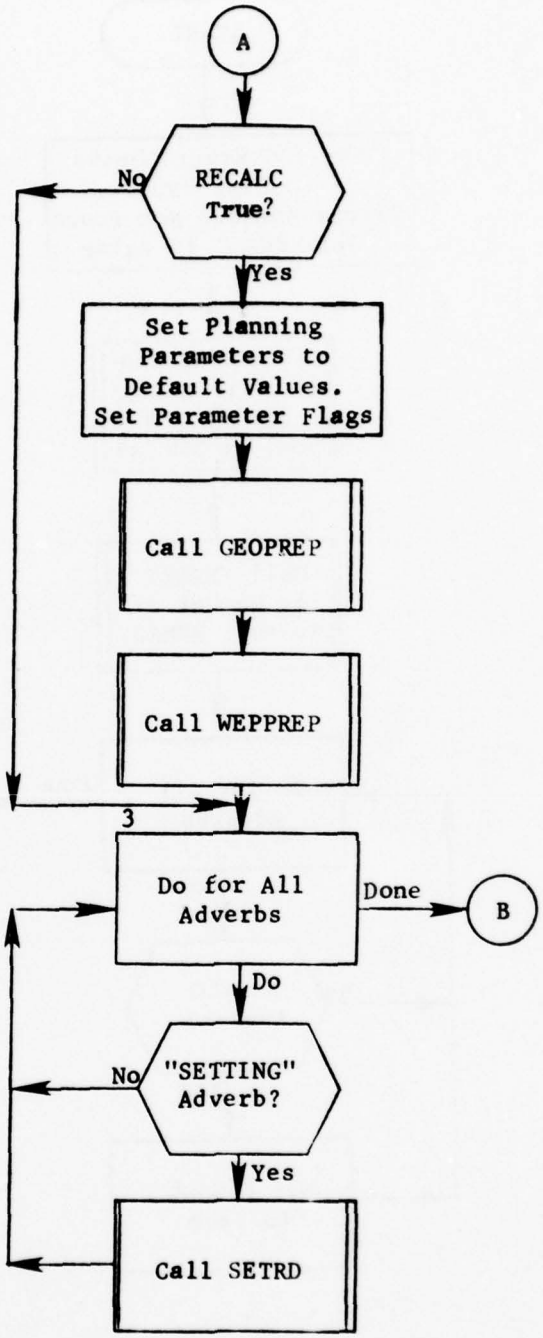


Figure 3. (Part 2 of 4)

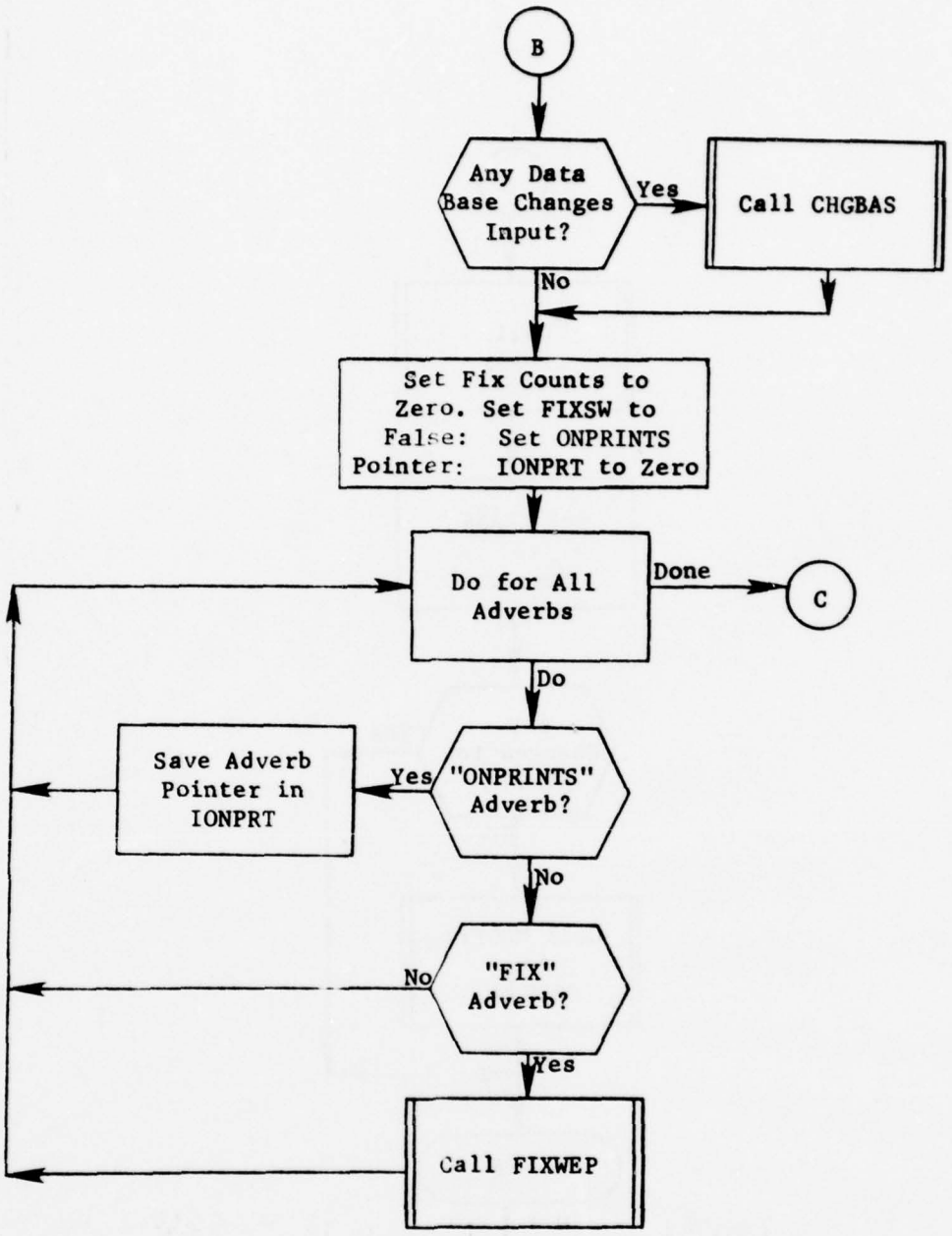


Figure 3. (Part 3 of 4)

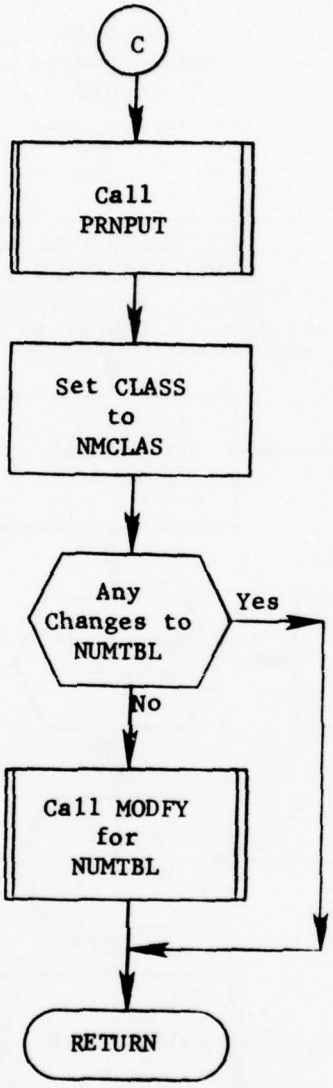


Figure 3. (Part 4 of 4)

2.8 Subroutine CHGBAS

PURPOSE: To make requested changes to target base parameters

ENTRY POINTS: CHGBAS

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C15, C30, SETCOM

SUBROUTINES CALLED: DIRECT, HDFND, HEAD, MODFY, NEXTTT, RETRV

CALLED BY: ENTMOD (of PREPALOC)

Method:

The target list is examined and each simple target or element of a complex is compared to the list of requested changes to see if any of them apply. When a particular condition is found to apply to a target the attribute involved is altered as per the change request. Change requests have been placed in order by SETRD such that for any given target only the first request to alter a particular attribute is honored.

As the elements of a complex are processed the subroutine keeps track of the overall effects on the complex of any changes and updates the complex the complex record as well when all of its elements have been processed. The effect of any changes to target values is noted throughout the cycle of the target list and, when the first cycle is complete, a second cycle takes place to renormalize all data base values.

Subroutine CHGBAS is illustrated in figure 4.

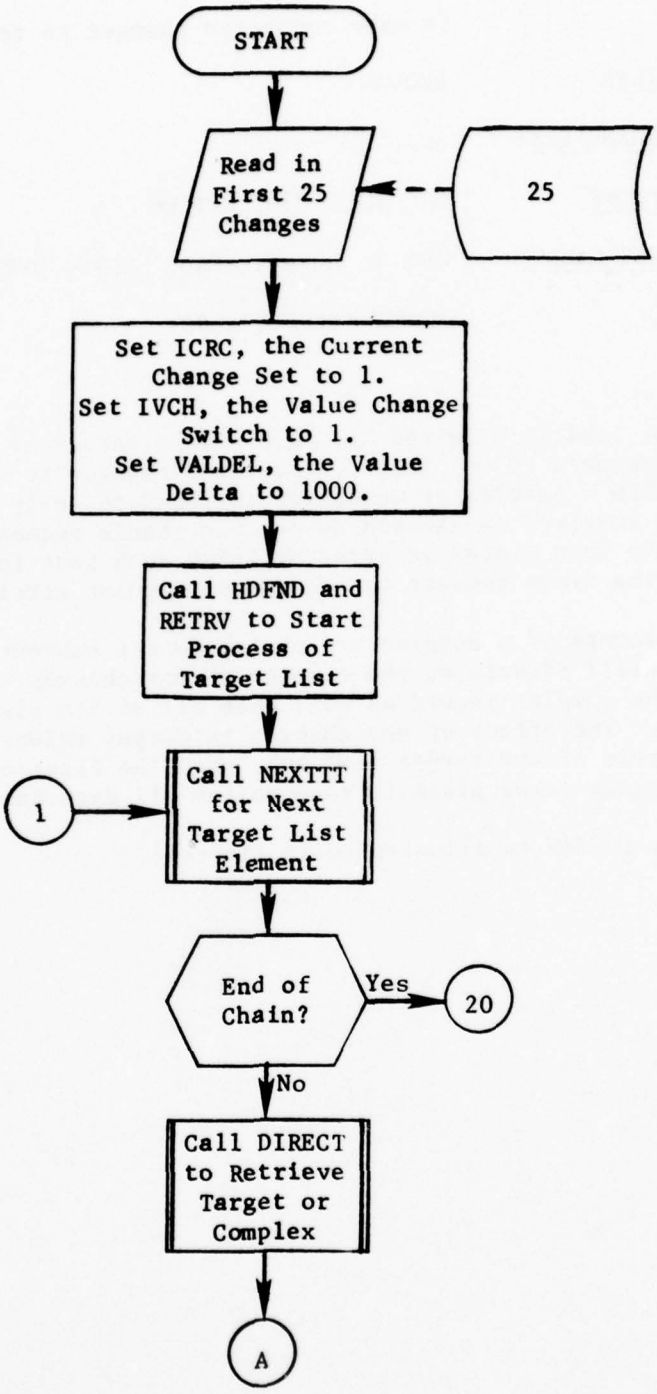


Figure 4. Subroutine CHGBAS (Part 1 of 9)

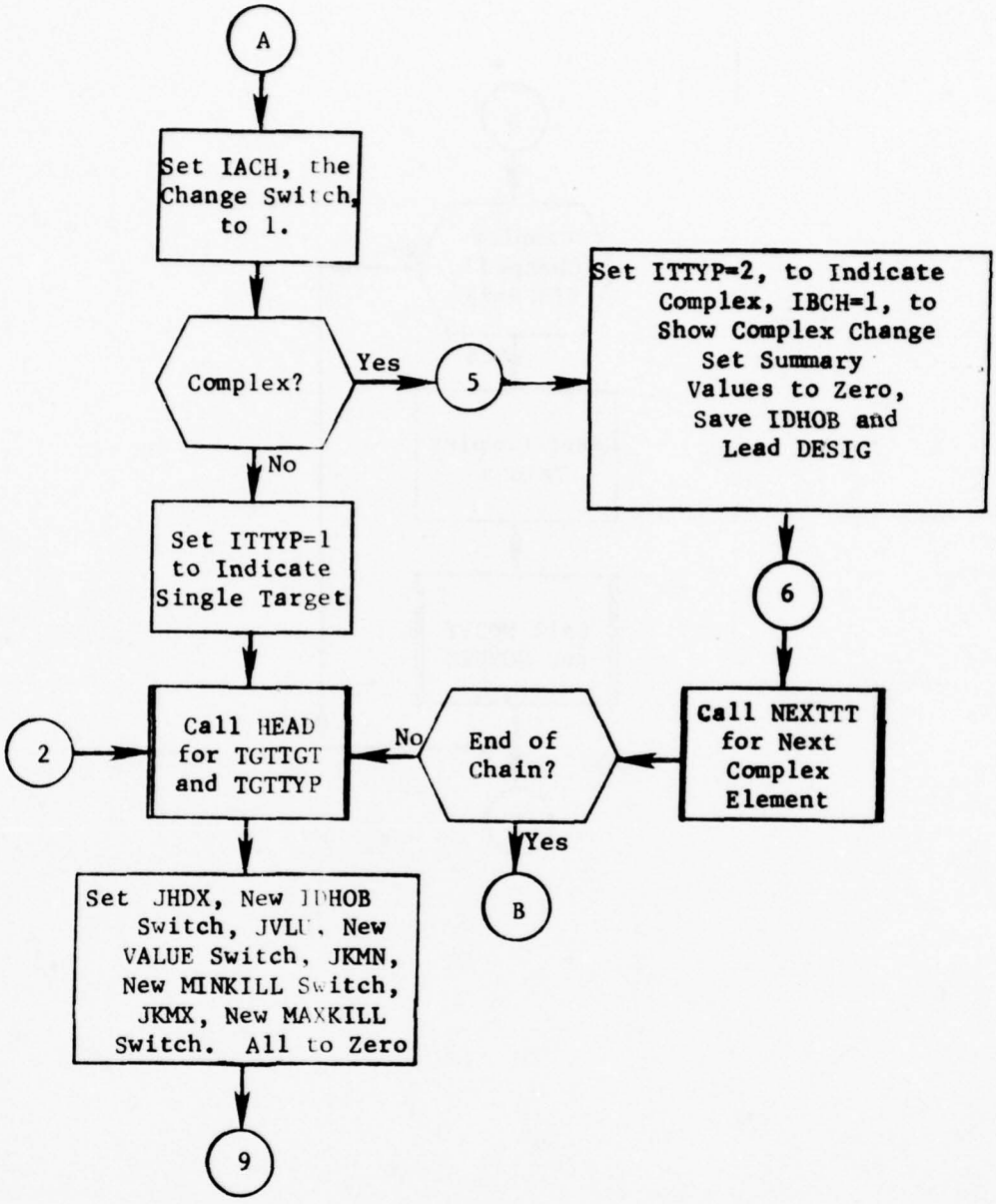


Figure 4. (Part 2 of 9)

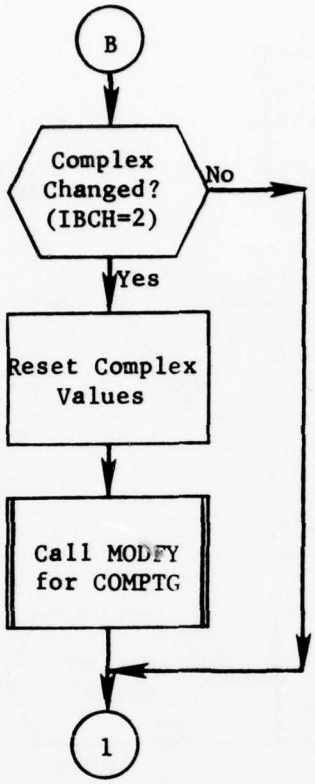


Figure 4. (Part 3 of 9)

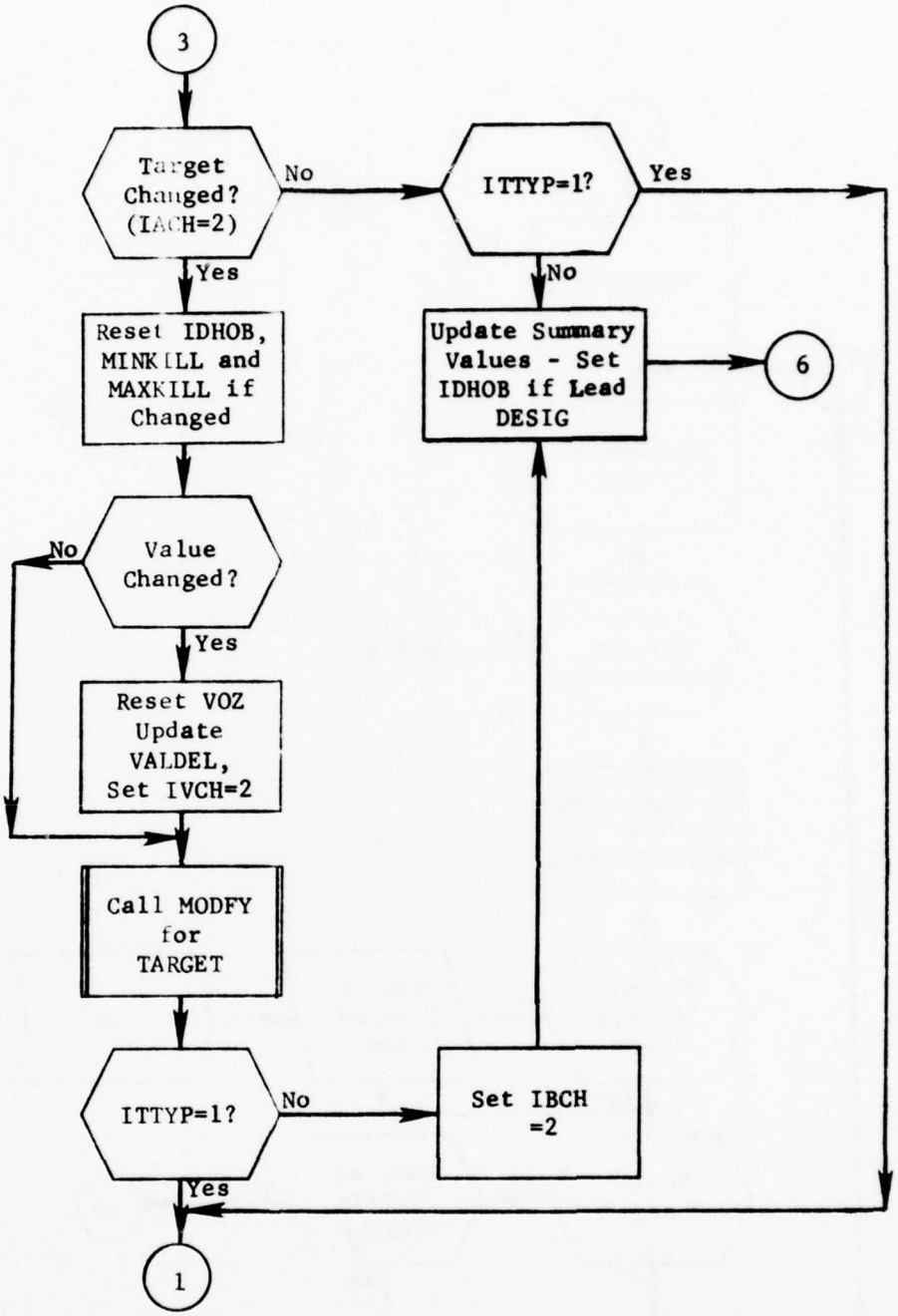


Figure 4. (Part 4 of 9)

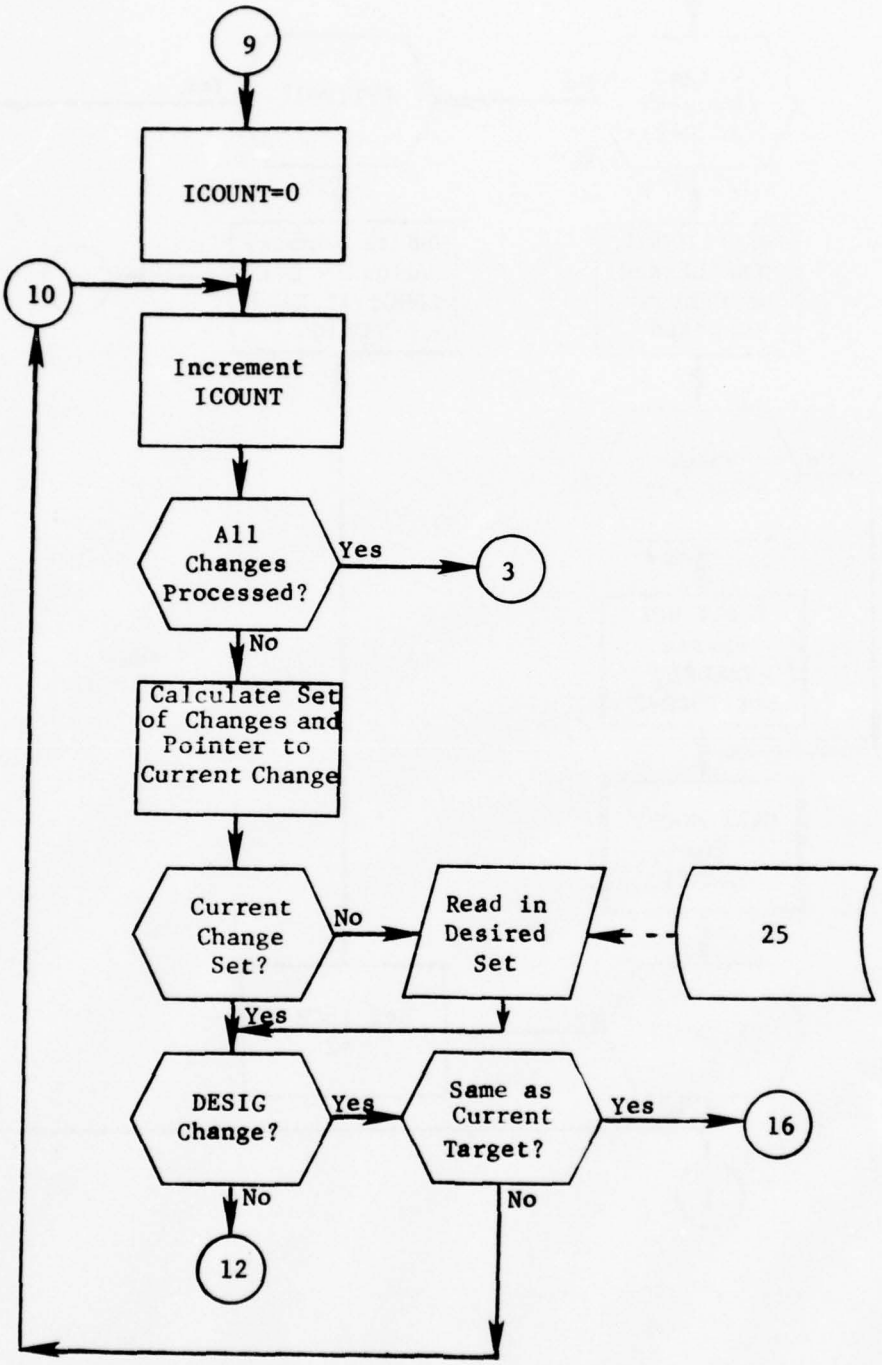


Figure 4. (Part 5 of 9)

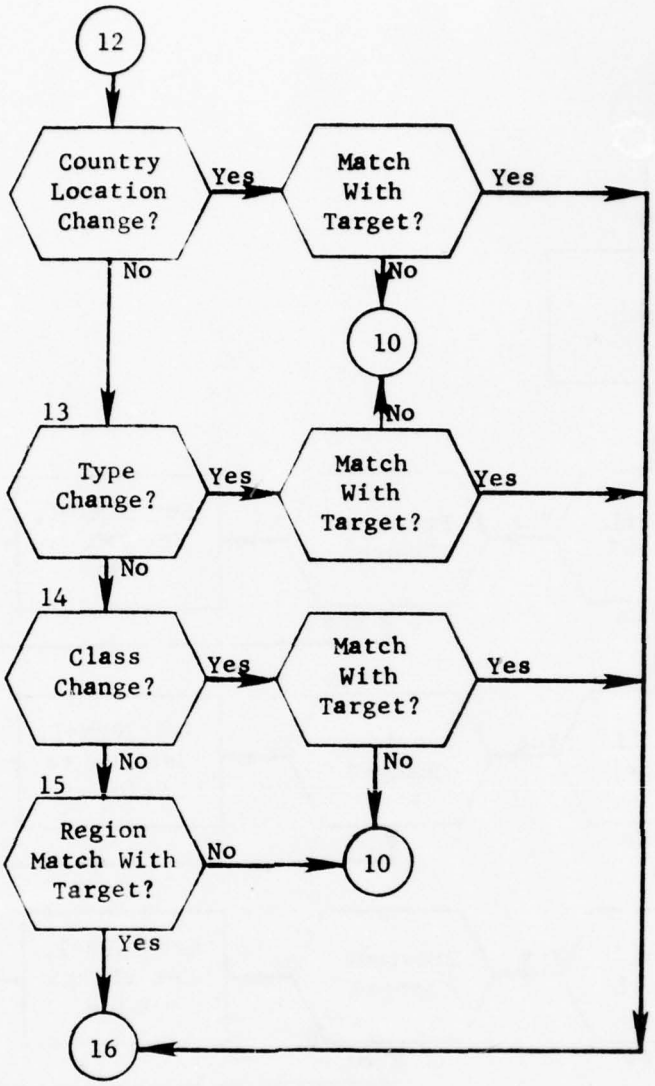


Figure 4. (Part 6 of 9)

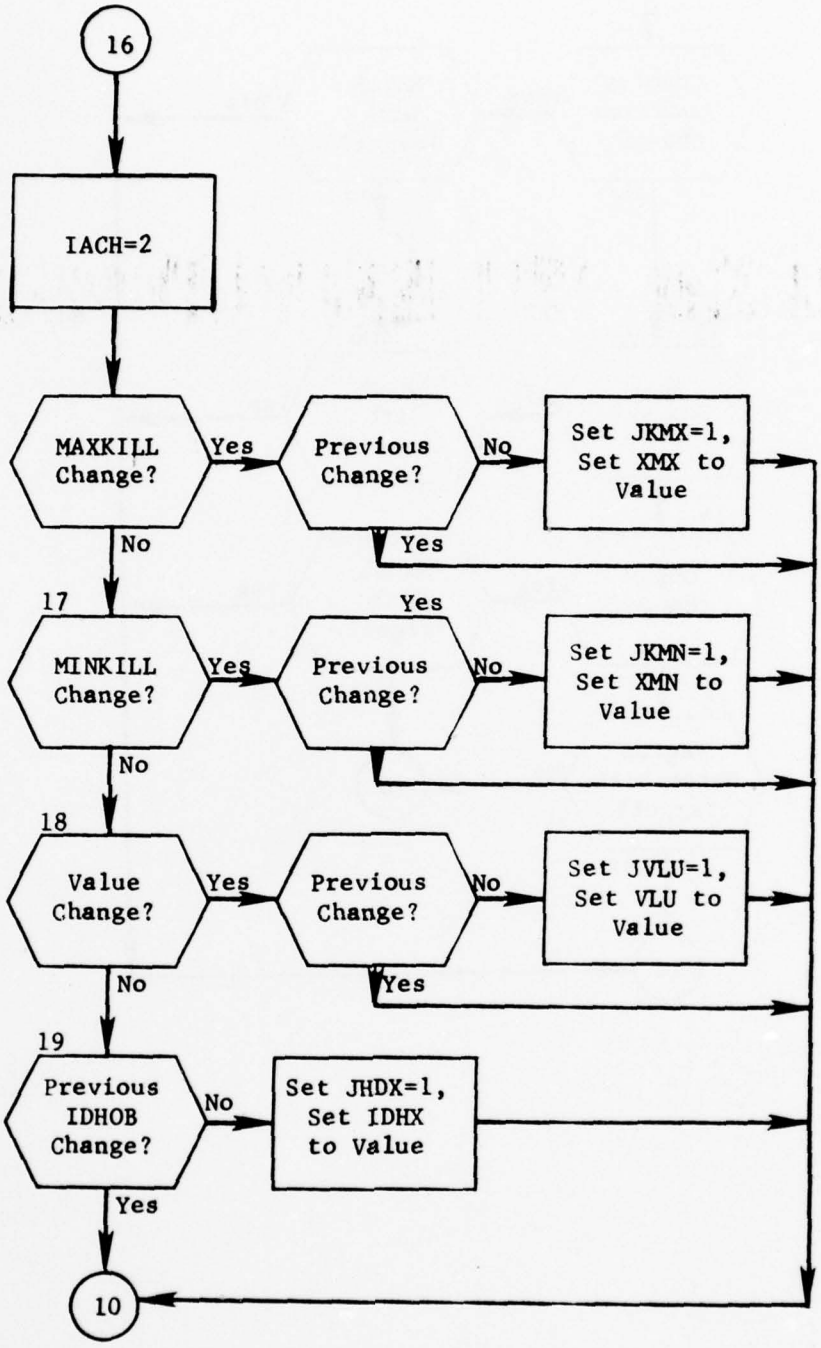


Figure 4. (Part 7 of 9)

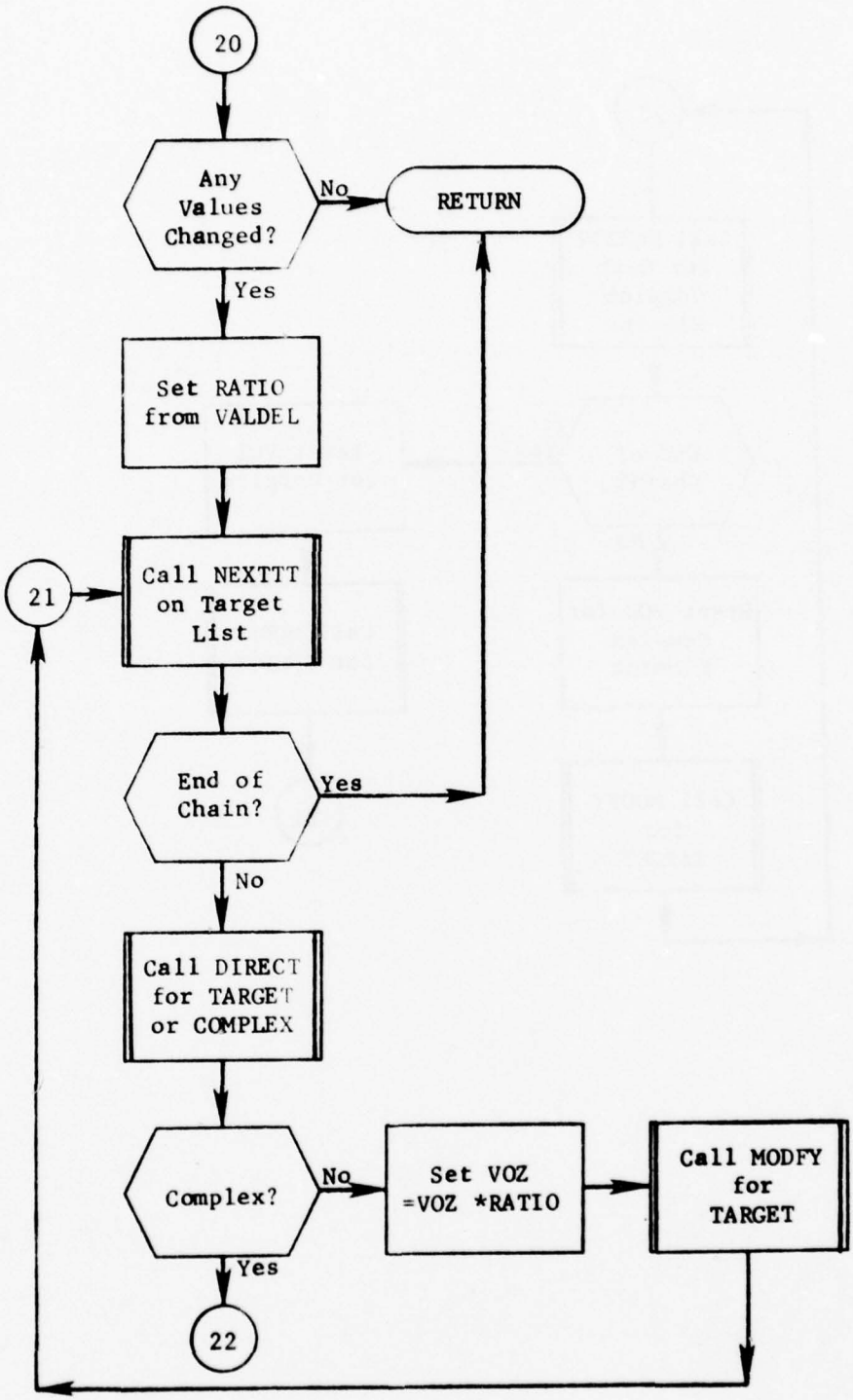


Figure 4. (Part 8 of 9)

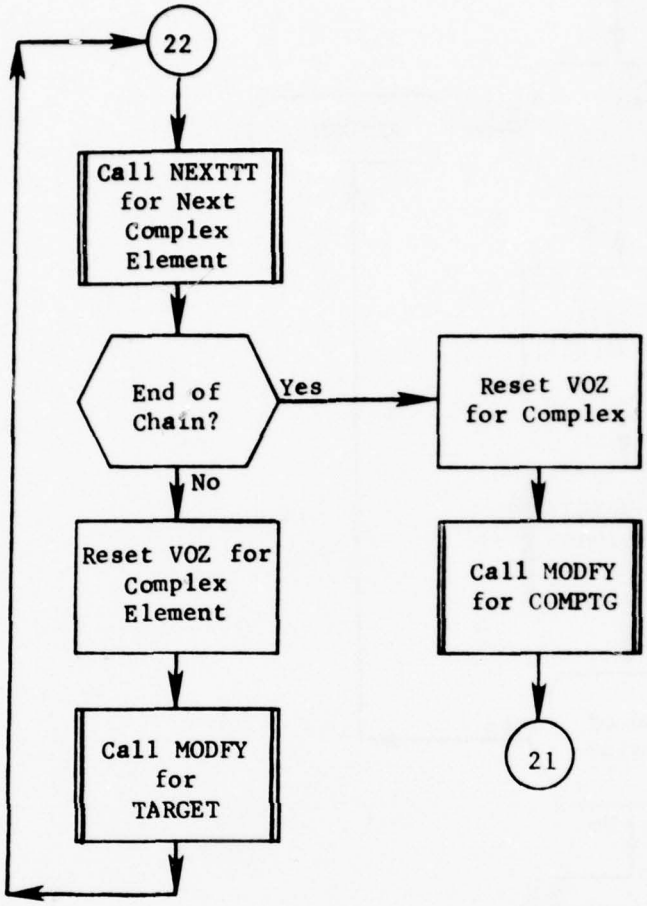


Figure 4. (Part 9 of 9)

2.9 Subroutine FIXWEP

PURPOSE: To create fixed weapon assignments

ENTRY POINTS: FIXWEP

FORMAL PARAMETERS: ICPT - index of FIX clause

COMMON BLOCKS: C10, C15, C30, IFXREQ, OOPS, TSTUFF, WEPCOM, ZEES

SUBROUTINES CALLED: DIRECT, DISTF, HDFND, HEAD, INSGET, MODFY, NEXTTT, RETRV, STORE, TOFM, WEPIN

CALLED BY: ENTMOD (of PREPALOC)

Method:

The first step in the FIXWEP process is to retrieve the target list header (TARNUM) and initialize break-point tables. These break-point tables are used to speed retrieval of records on the LISTXX chain. By saving the direct reference code of up to 500 records evenly spaced along the chain, any record may be retrieved by beginning the search at the nearest break-point record that precedes it.

The remainder of the process is driven by the FIX clause which caused the call. The FIX clause is viewed as a series of subclauses each of which contains a subject and one or more objects. The subject is a collection of attributes and each object is a collection of values for those attributes. As a subject is read, its component attributes are used to set up a group of switches to indicate the operations to be carried out for each object. The switches are:

ISRNG - 1 if one DESIG, 2 if two DESIGs
IASET - 1 if no set arrival time, 2 if arrival time is set

FIXWEP then reads in each related object and creates ASSIGN records as follows. Each object will set the following values:

IXGRP - the group number
ISTSAL - the selected salvo (0 if non selected)
AXRV - the selected arrival time (0 if IASET = 1)
INMA - the number of assignments (set by NUMALOC)
CDSG - the DESIG of the target

If there is a DESIG range (ISRNG = 2) the values of the range are placed in CDSG in order and the process carried out for each.

For each set of the above values, FIXWEP first checks to see if the current group (IGRP) is the same as IXGRP. If not, the current group is modified and the new group retrieved. The target whose DESIG is CDSG is not obtained and the corresponding TARCDE record found through use

of the break-point tables. Now the process loops INMA times. In each loop FIXWEP first checks to see if group resources are exceeded. If not, the process proceeds depending upon whether or not the group is salvoed. For a non-salvoed group FIXWEP simply stores a new ASSIGN record.

For a salvoed group the salvo is determined in one of three ways. If neither a salvo nor an arrival time has been specified, a convenient salvo is selected. If a salvo was specified, it is used. If an arrival time was specified, the time of flight is calculated and the salvo set to fulfill the desired time of arrival. In any case, FIXWEP now checks to see if a weapon is available in the chosen salvo. If so, the ASSIGN record is stored. If not, FIXWEP finds a salvo that is available and uses that salvo instead unless the desired salvo was specified. In the later case, FIXWEP will search all current assignments for the weapon group with the desired salvo and shift to the available salvo that was either assigned with no specification or, failing to find such, one whose arrival time was specified. The new assignment may then be made to the desired salvo.

Subroutine FIXWEP is illustrated in figure 5.

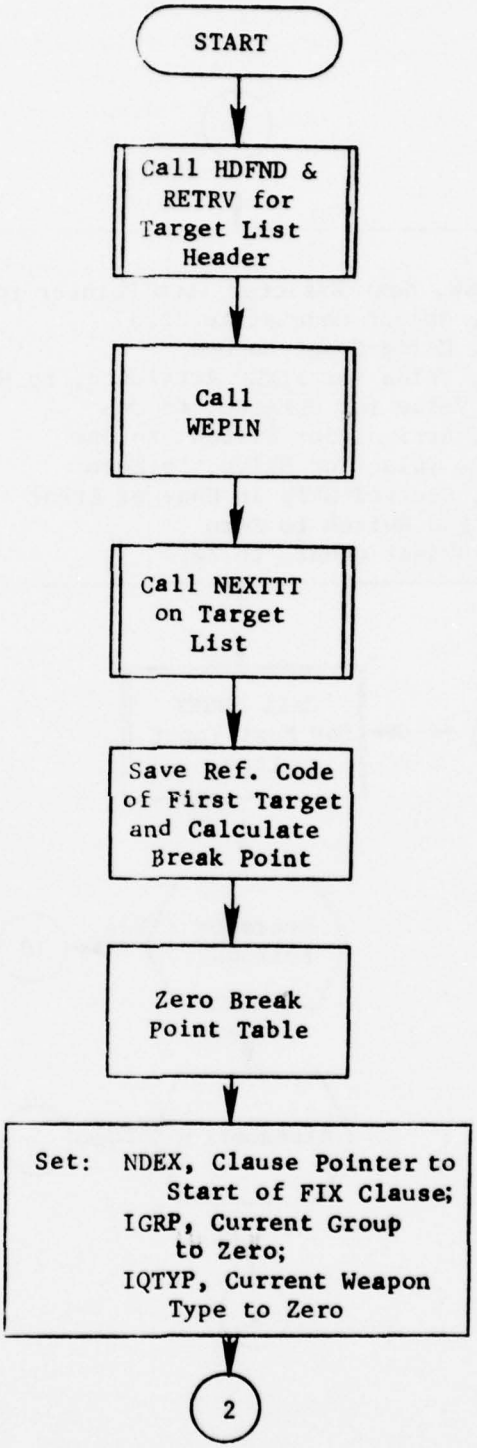


Figure 5. Subroutine FIXWEP (Part 1 of 23)

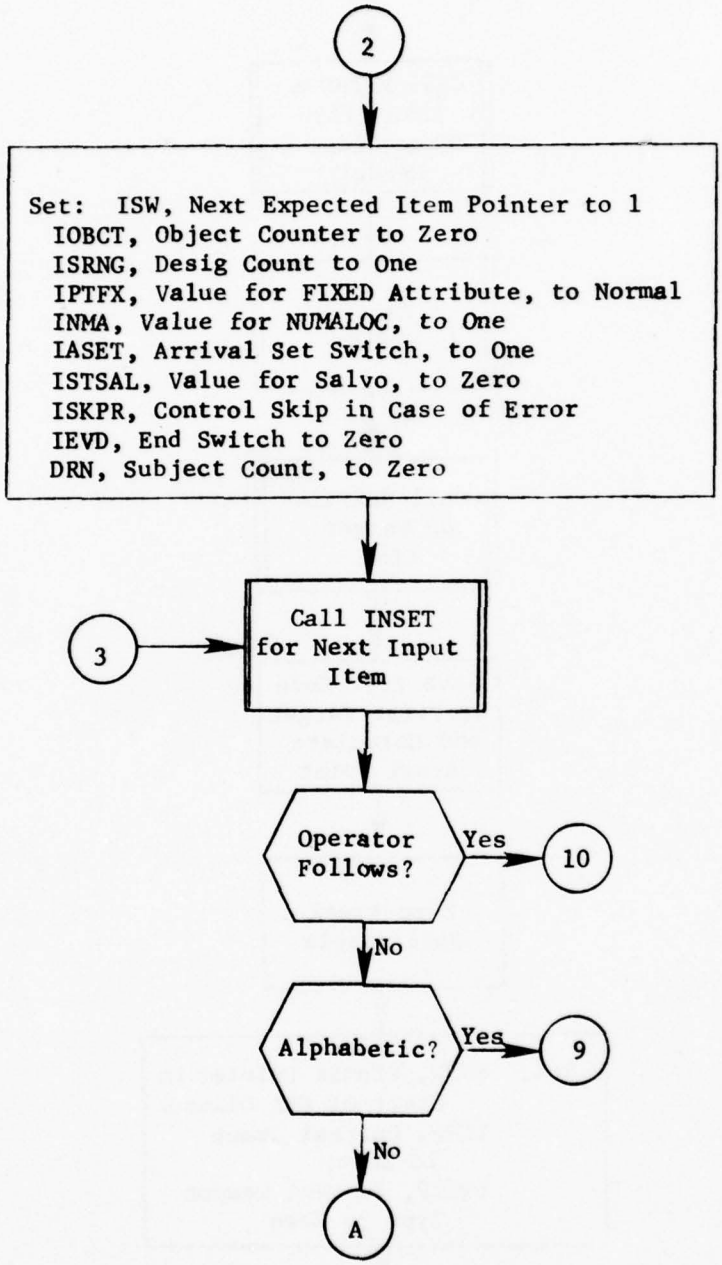


Figure 5. (Part 2 of 23)

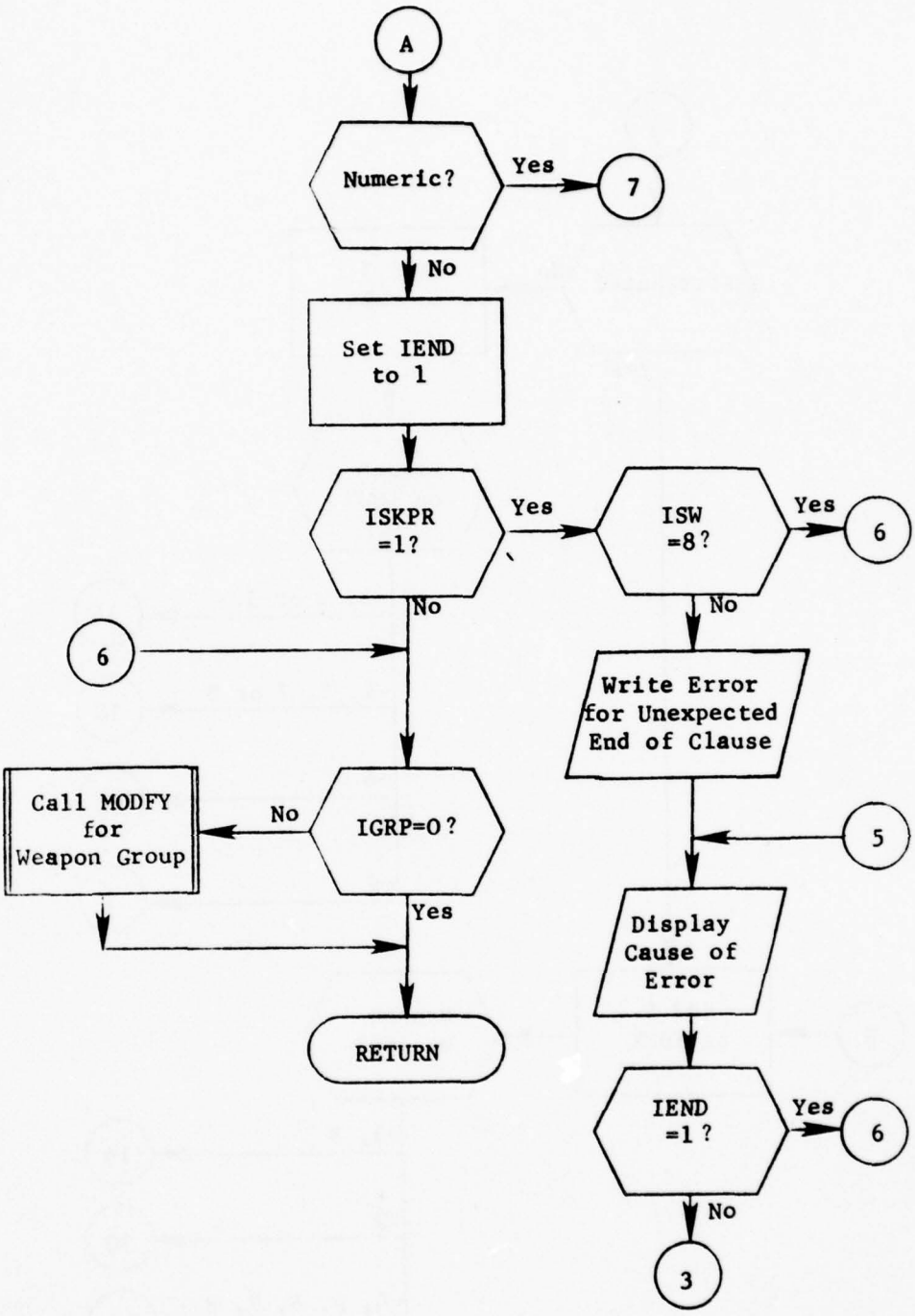


Figure 5. (Part 3 of 23)

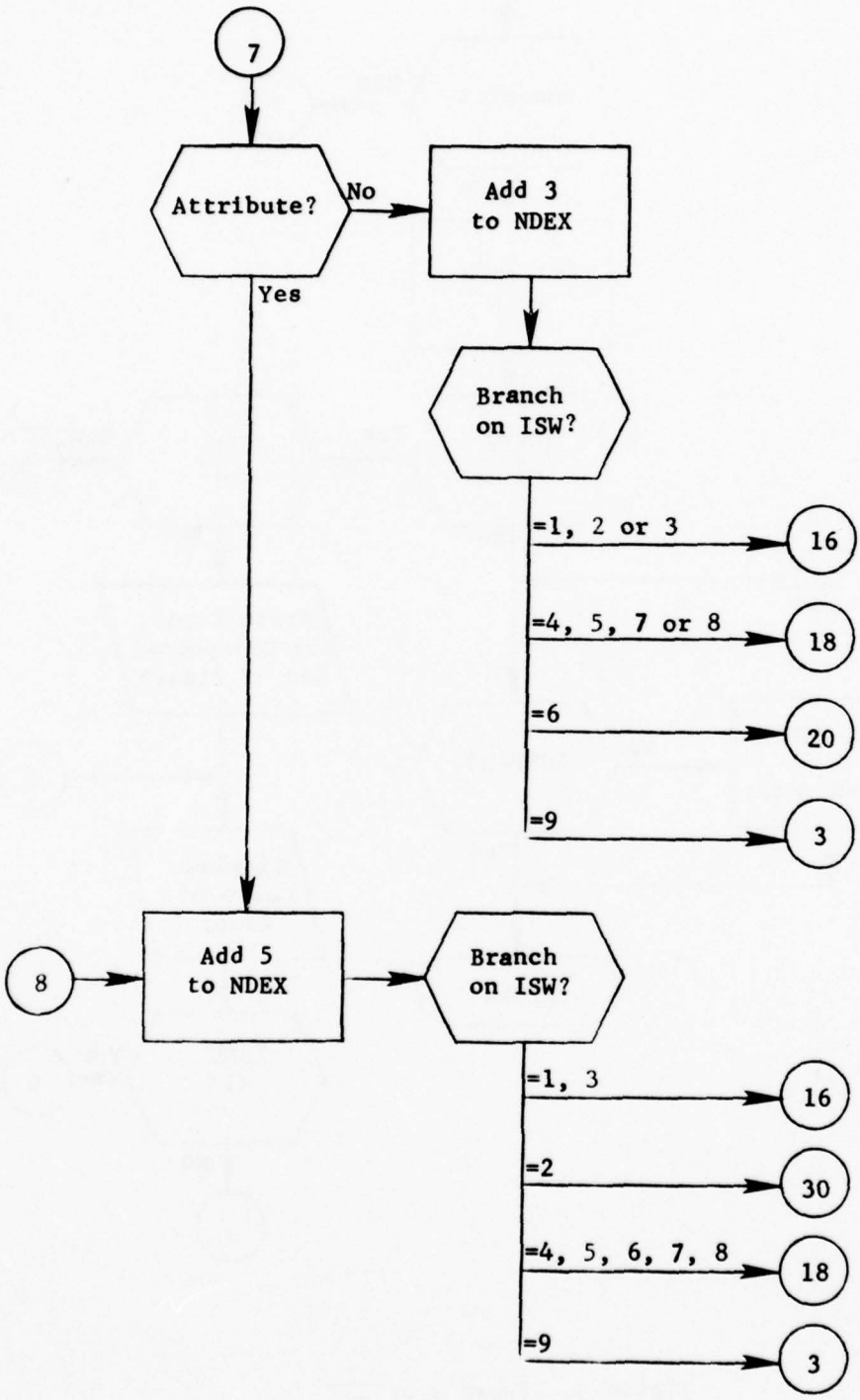


Figure 5. (Part 4 of 23)

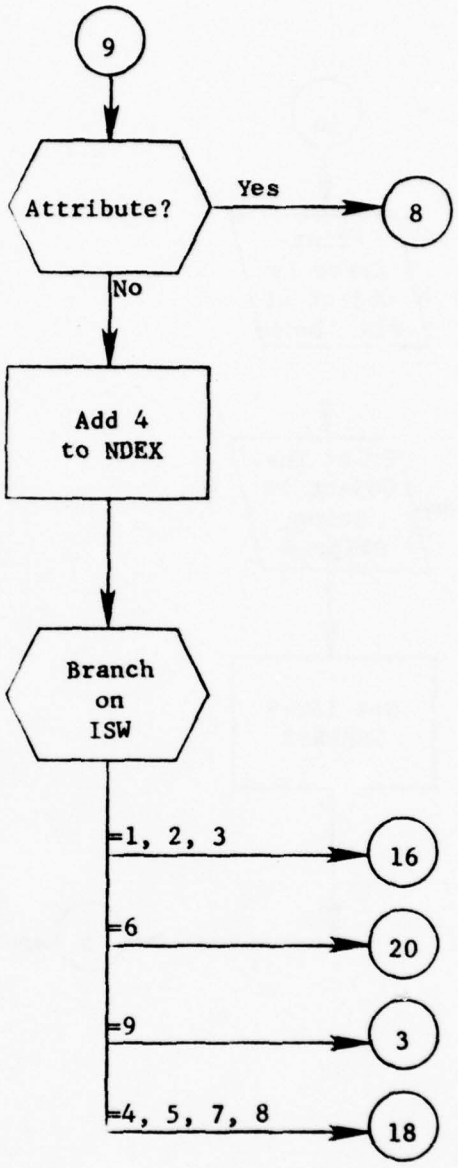


Figure 5. (Part 5 of 23)

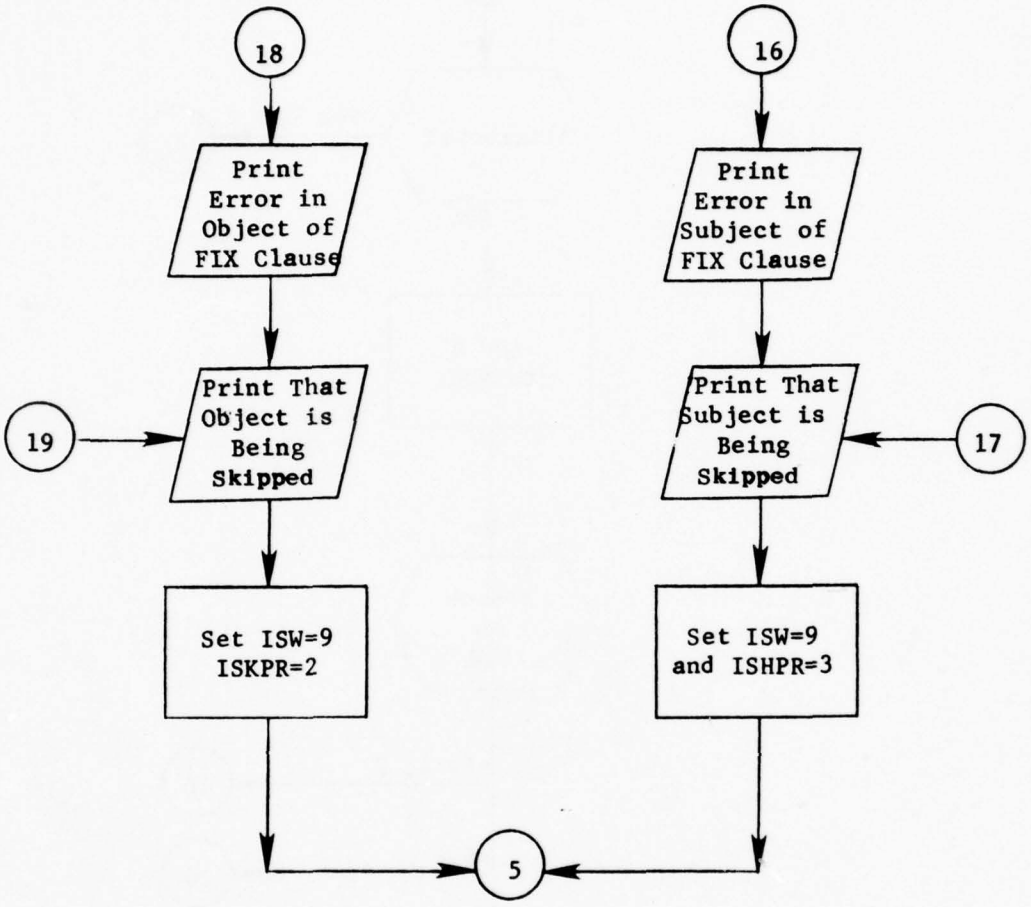


Figure 5. (Part 6 of 23)

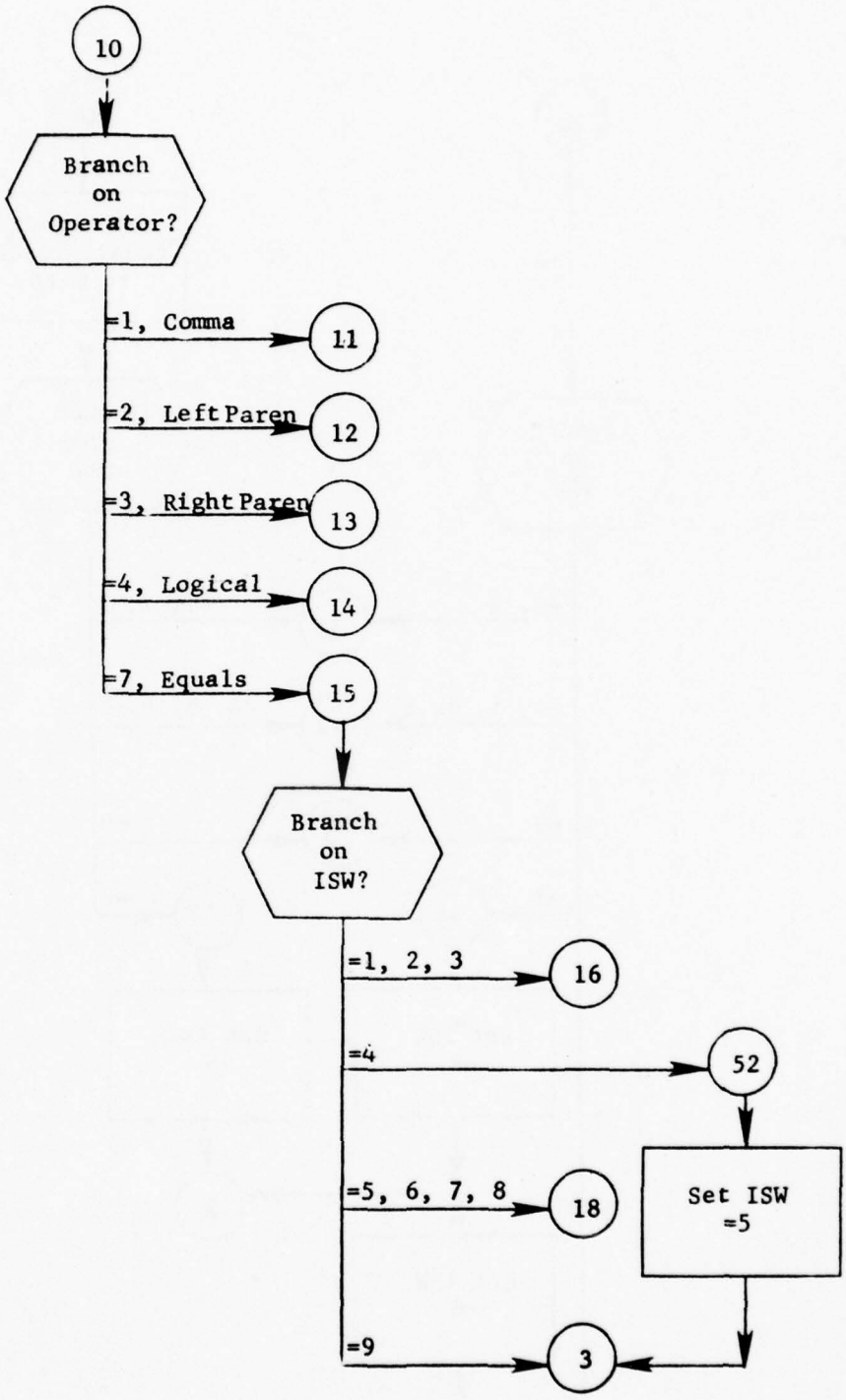


Figure 5. (Part 7 of 23)

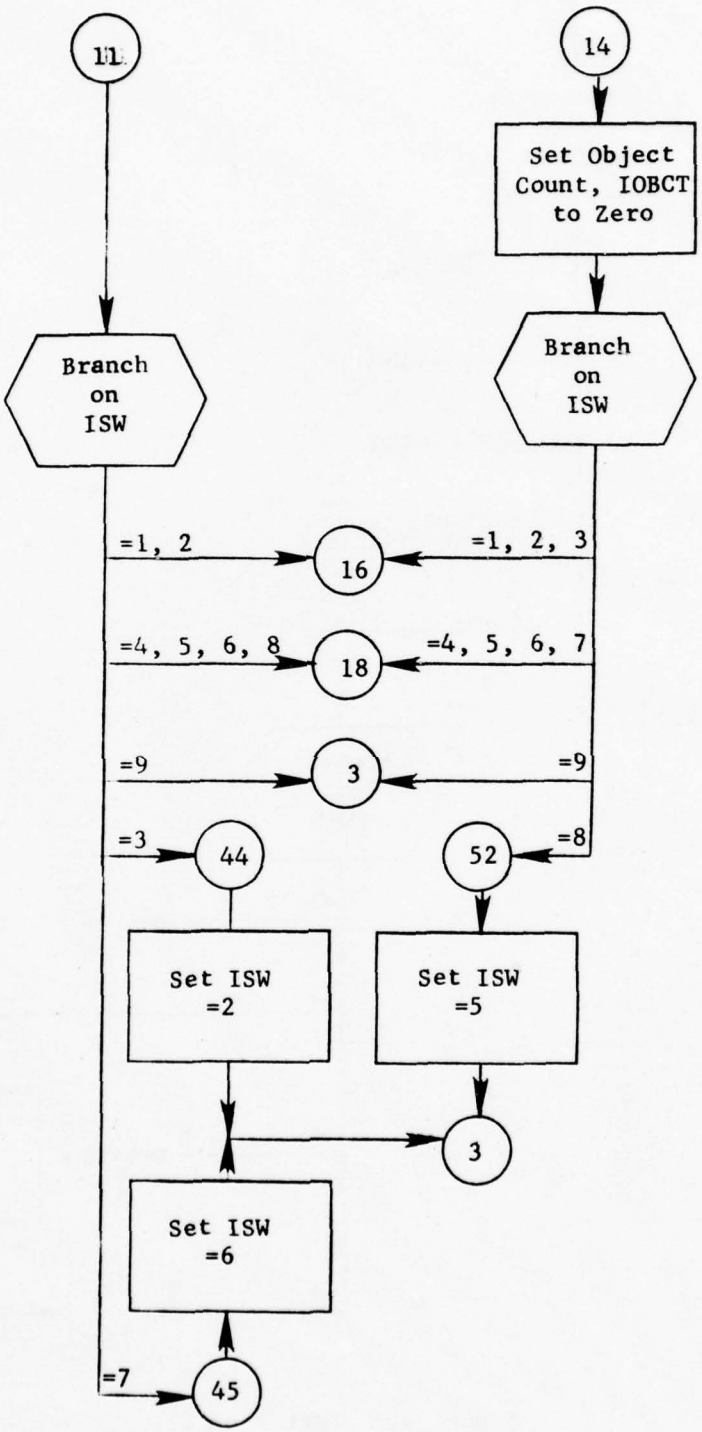


Figure 5. (Part 8 of 23)

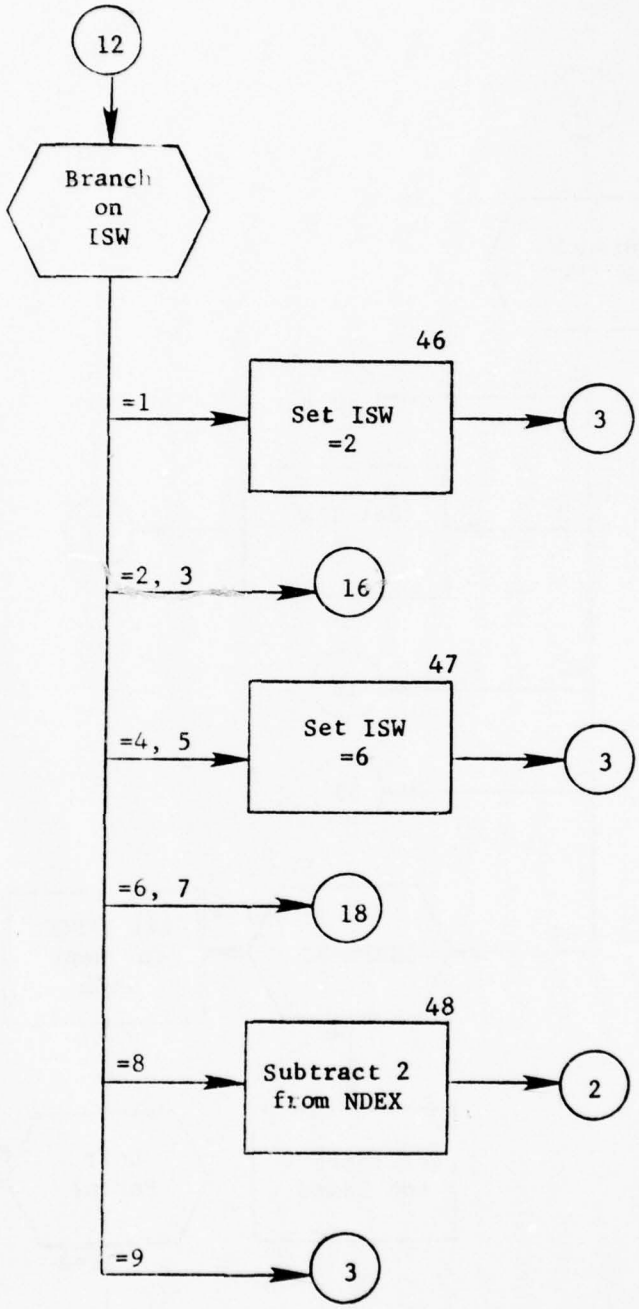


Figure 5. (Part 9 of 23)

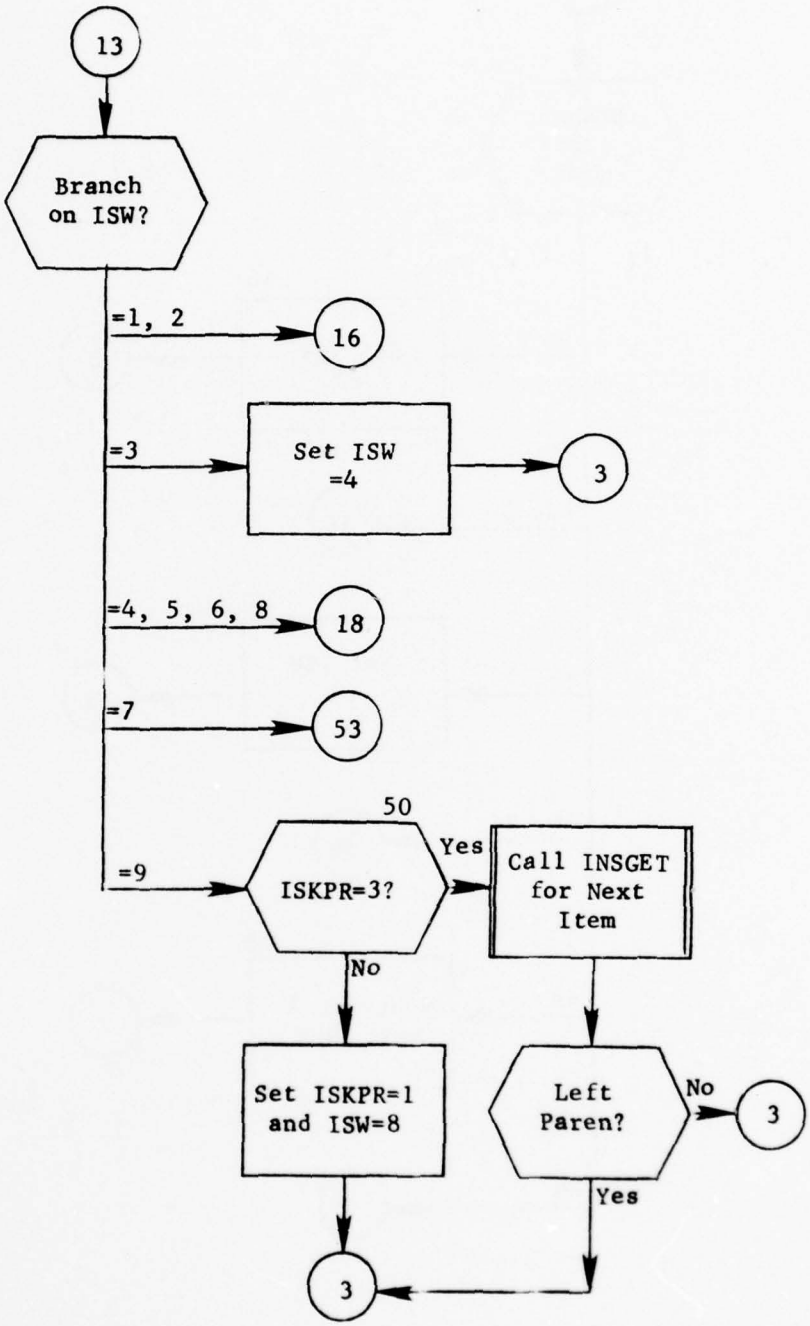


Figure 5. (Part 10 of 23)

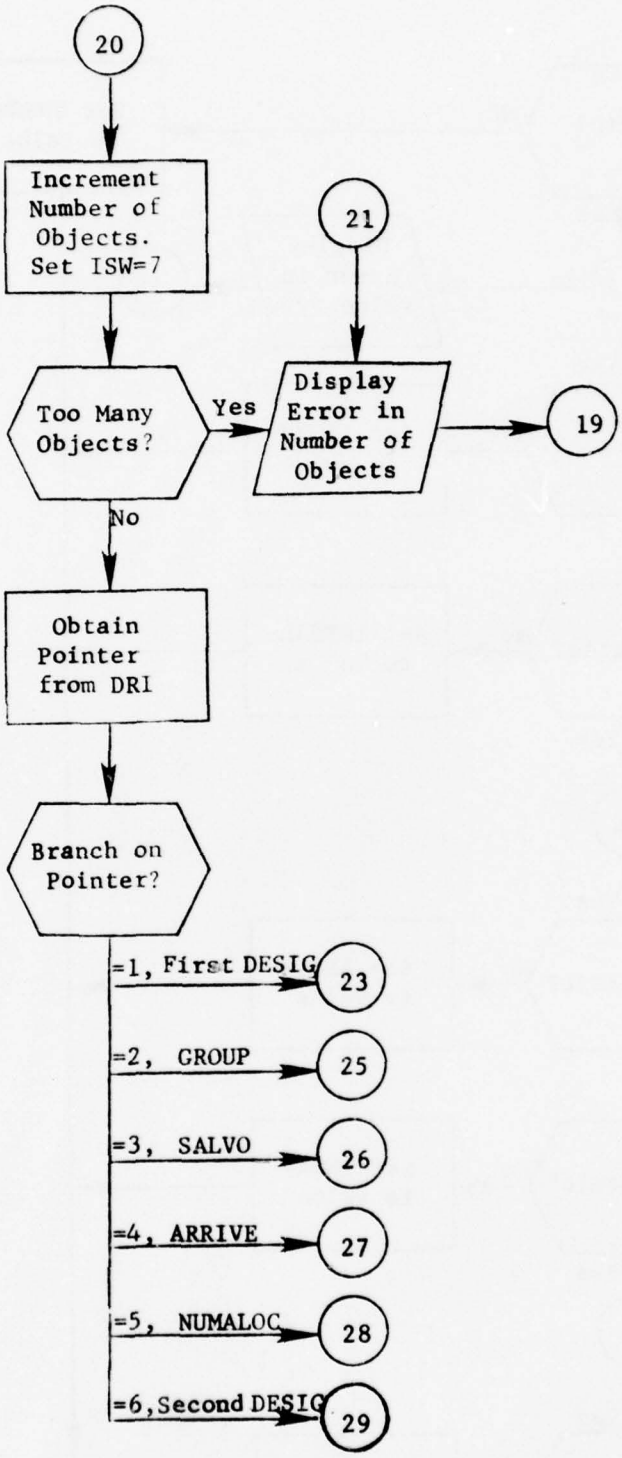


Figure 5. (Part 11 of 23)

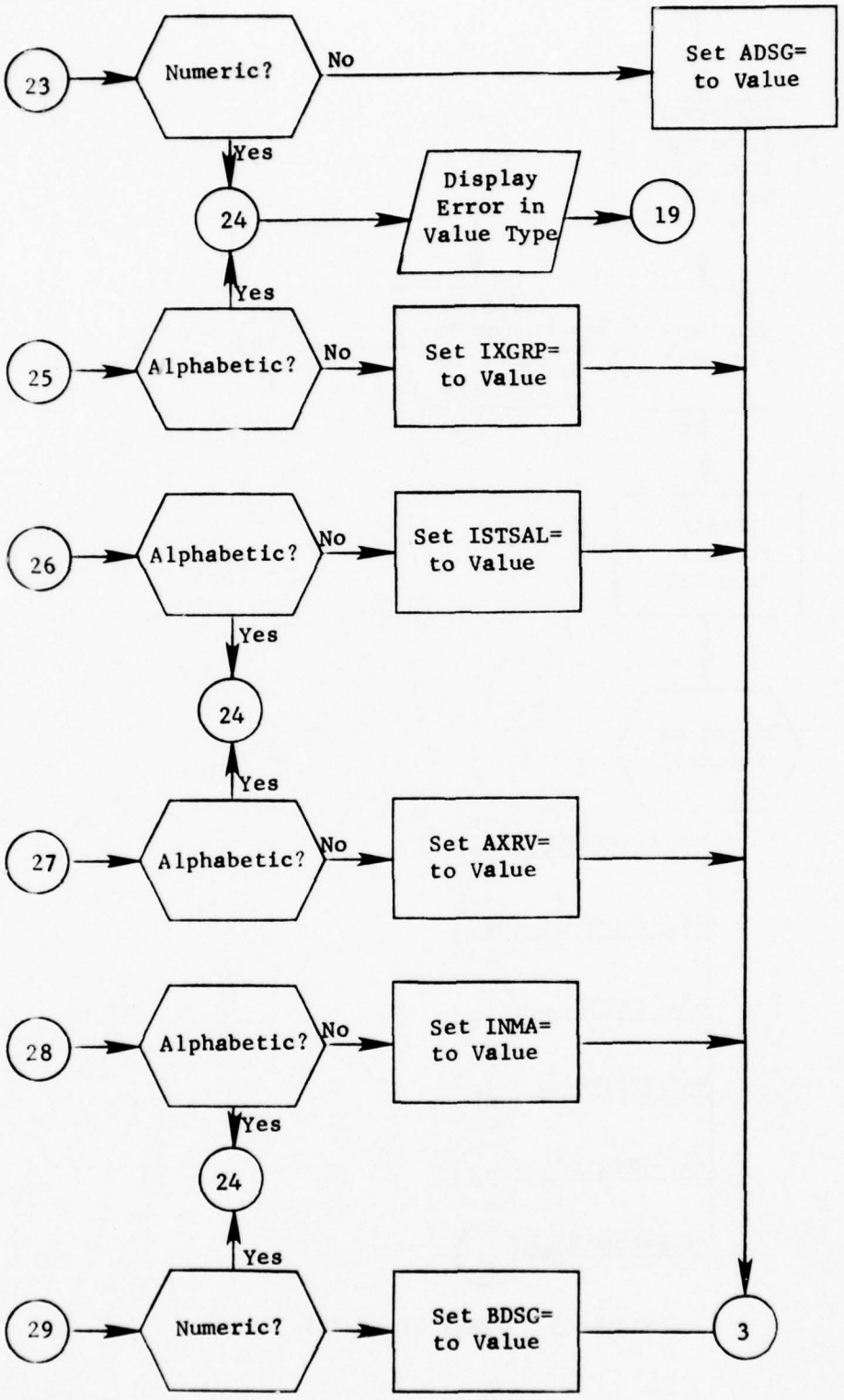


Figure 5. (Part 12 of 23)

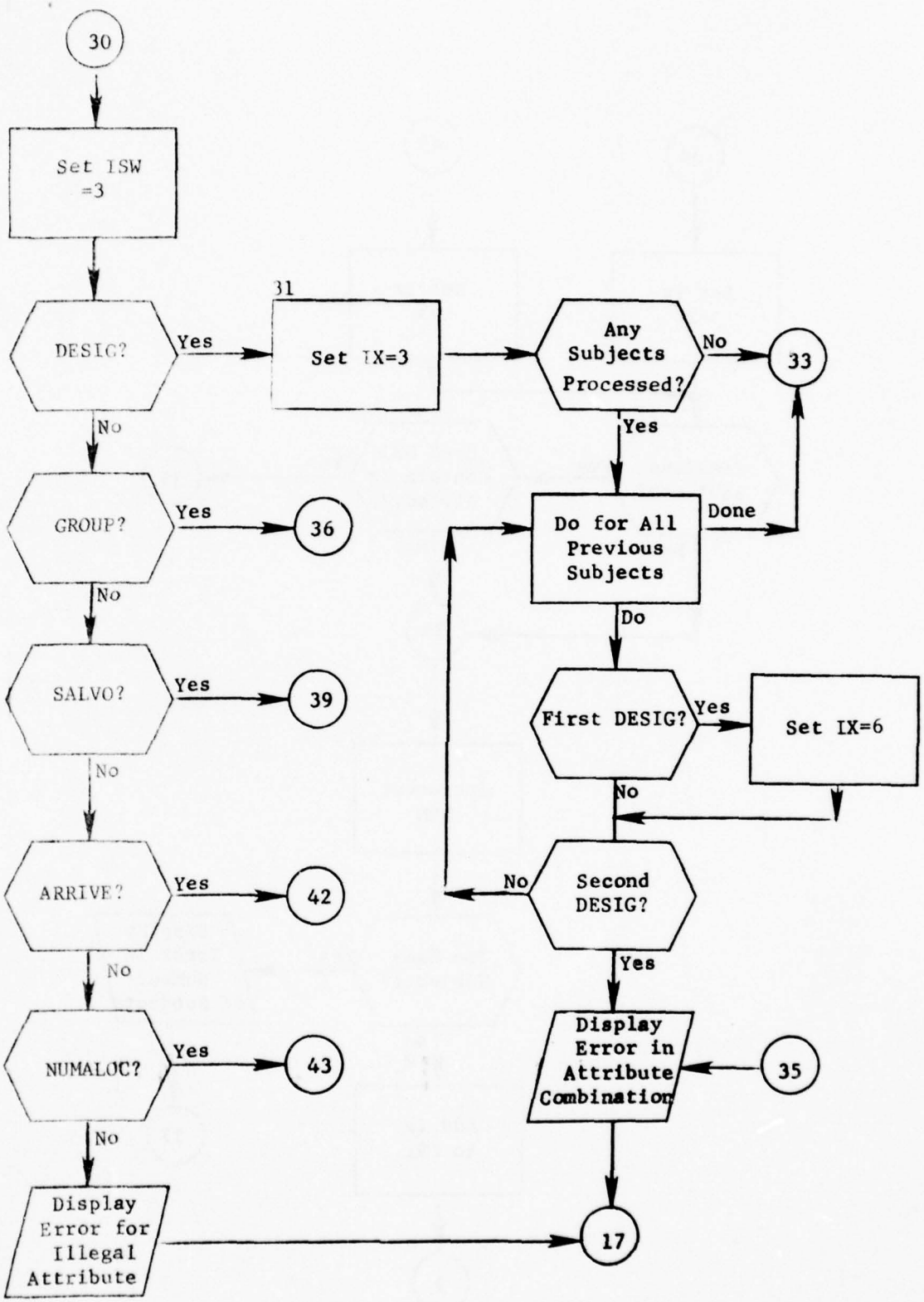


Figure 5. (Part 13 of 23)

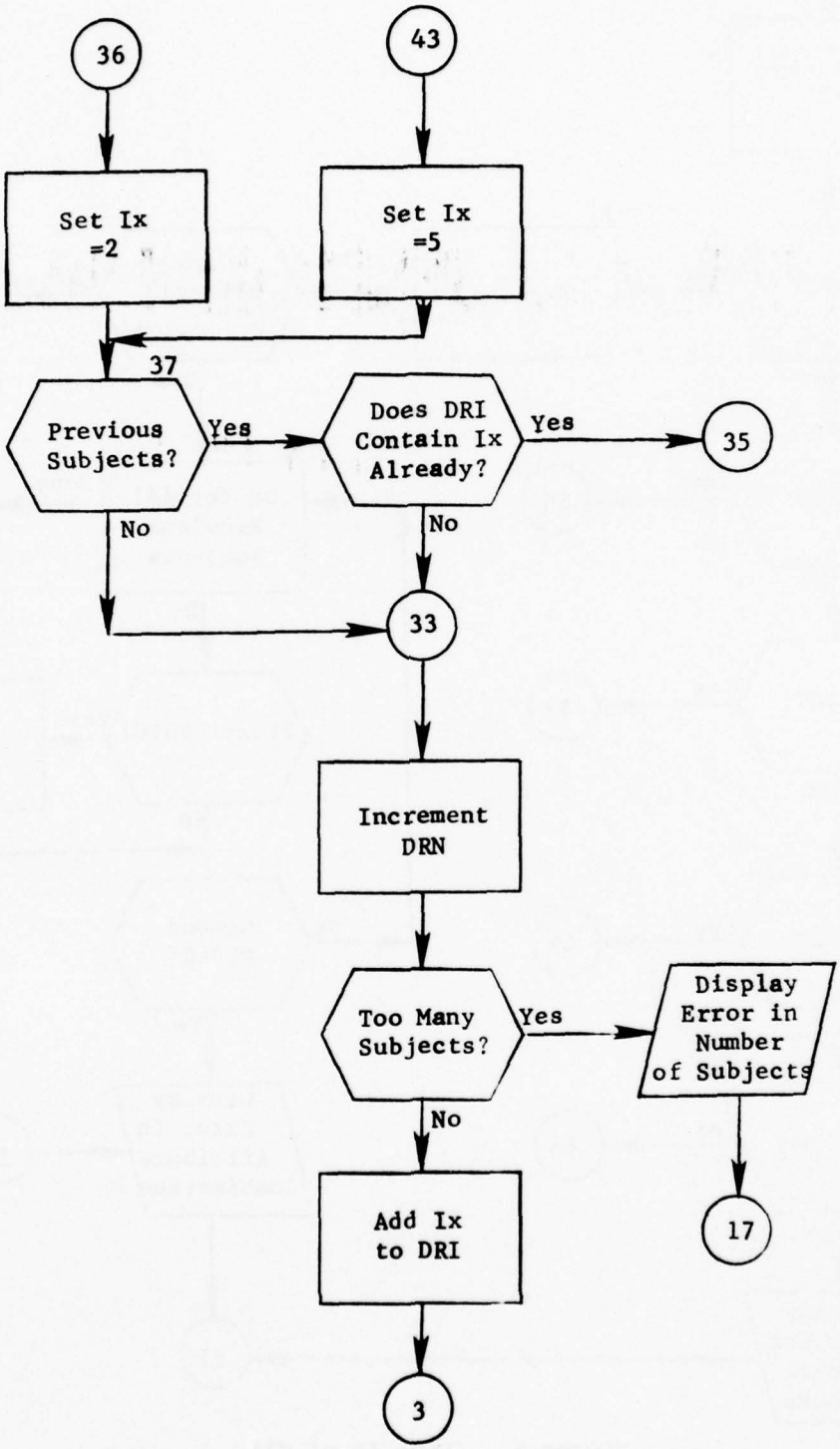


Figure 5. (Part 14 of 23)

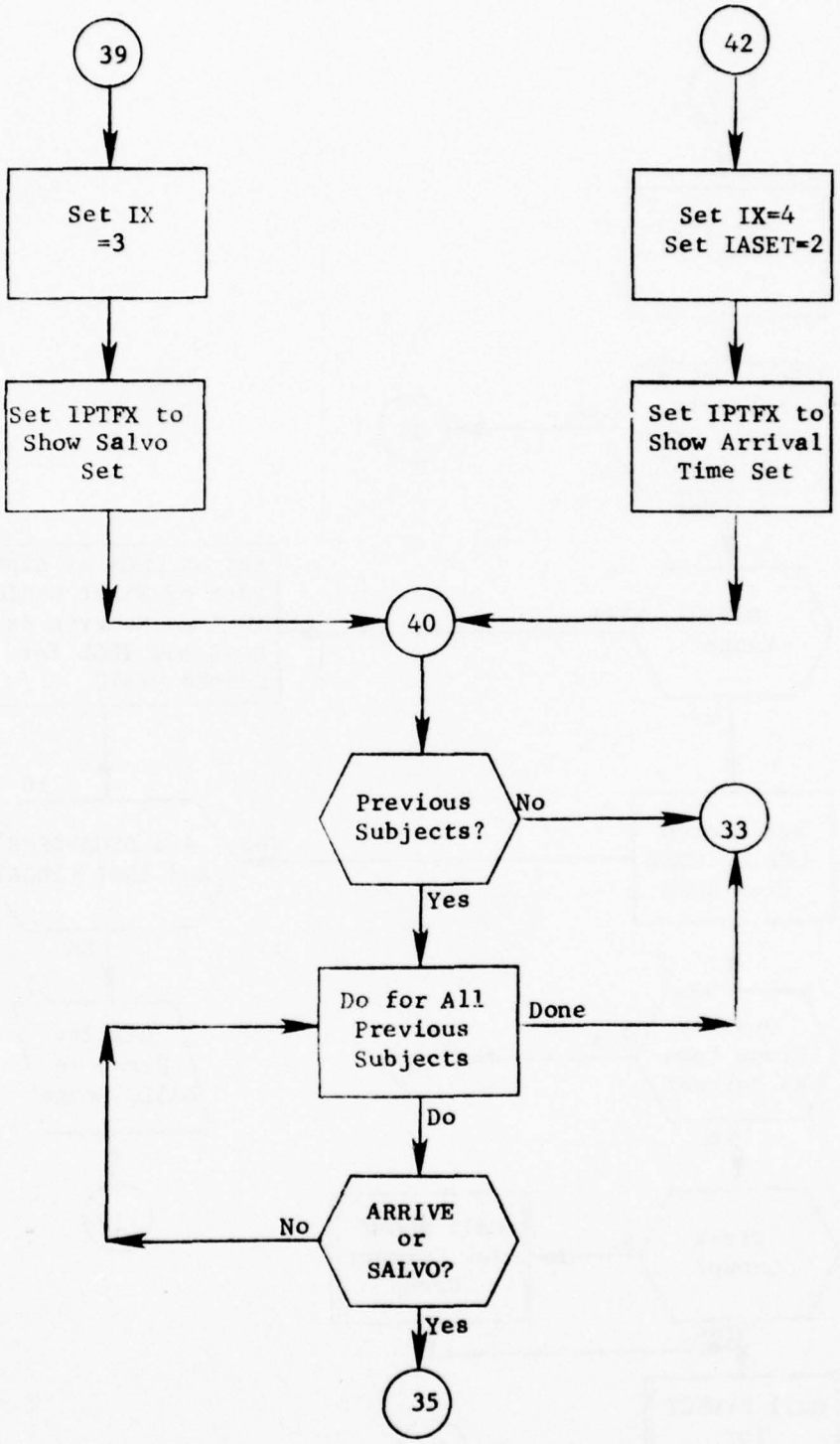


Figure 5. (Part 15 of 23)

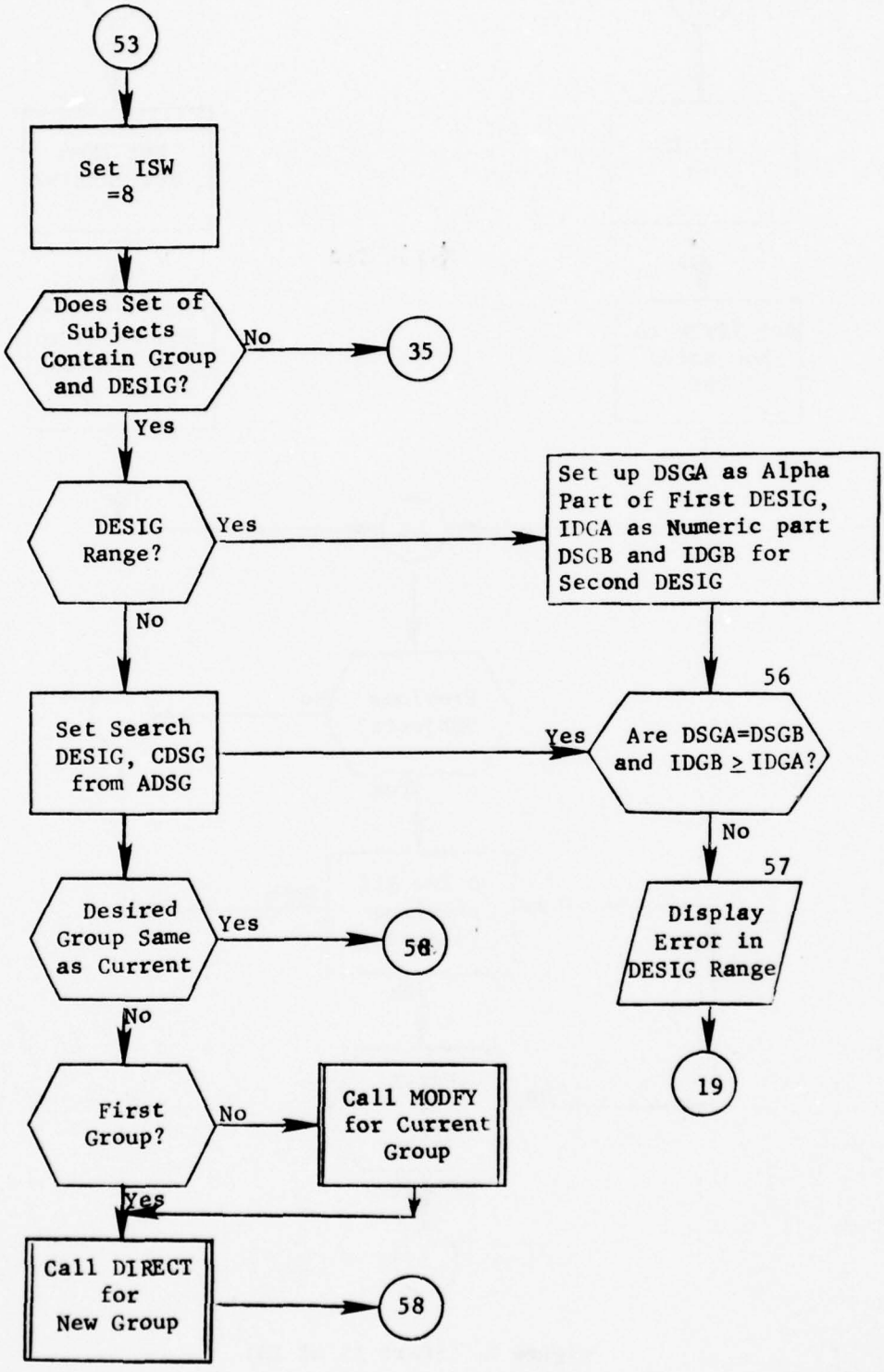


Figure 5. (Part 16 of 23)

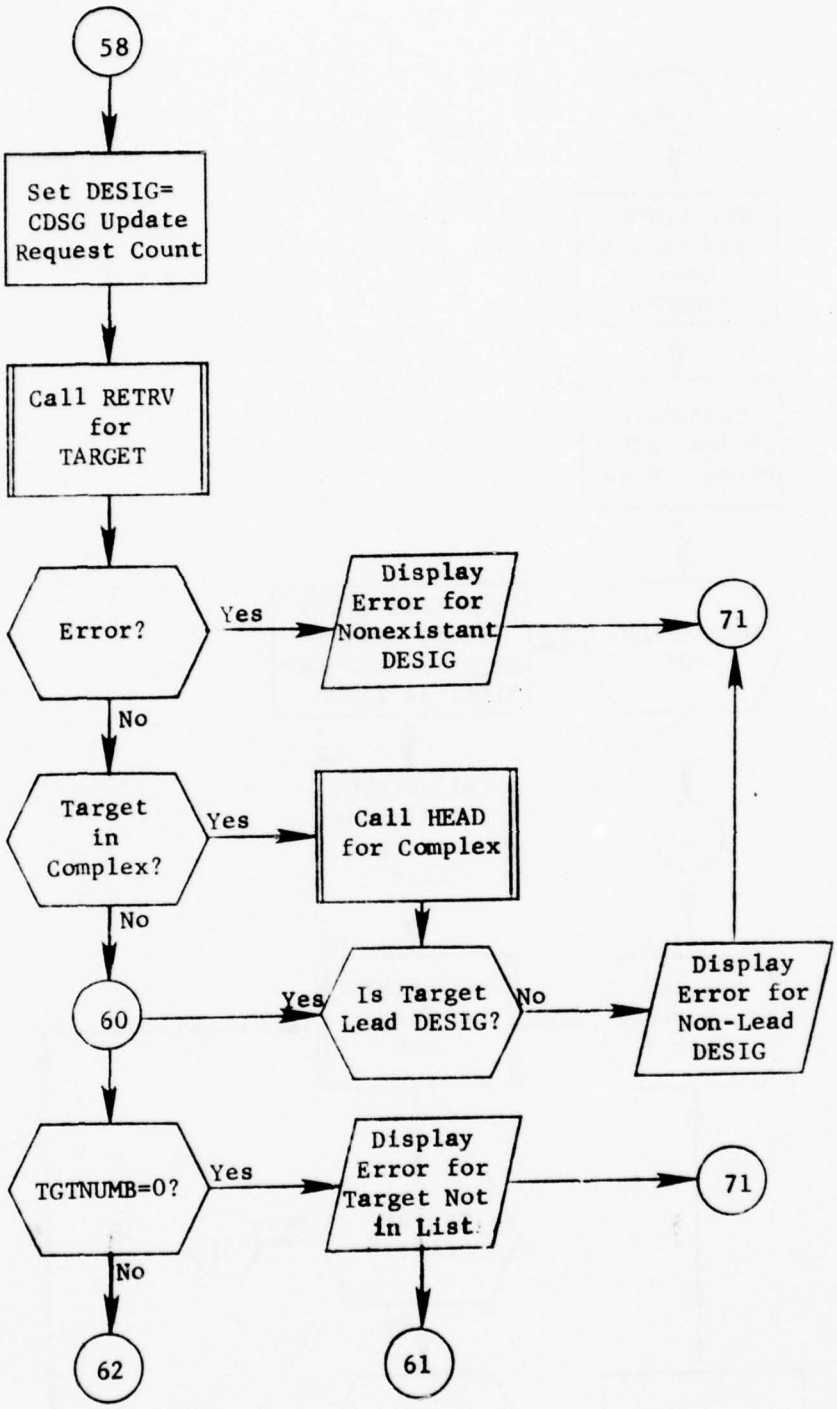


Figure 5. (Part 17 of 23)

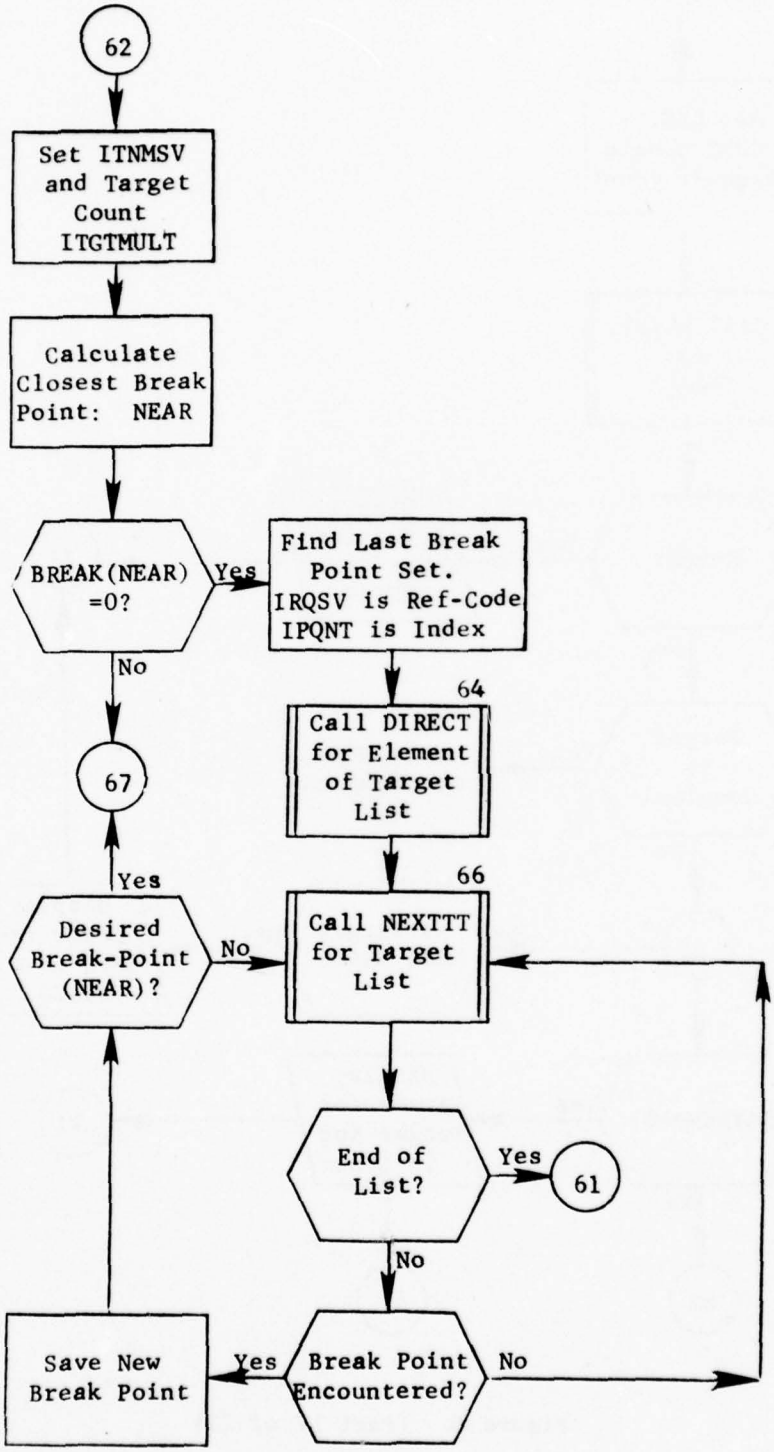


Figure 5. (Part 18 of 23)

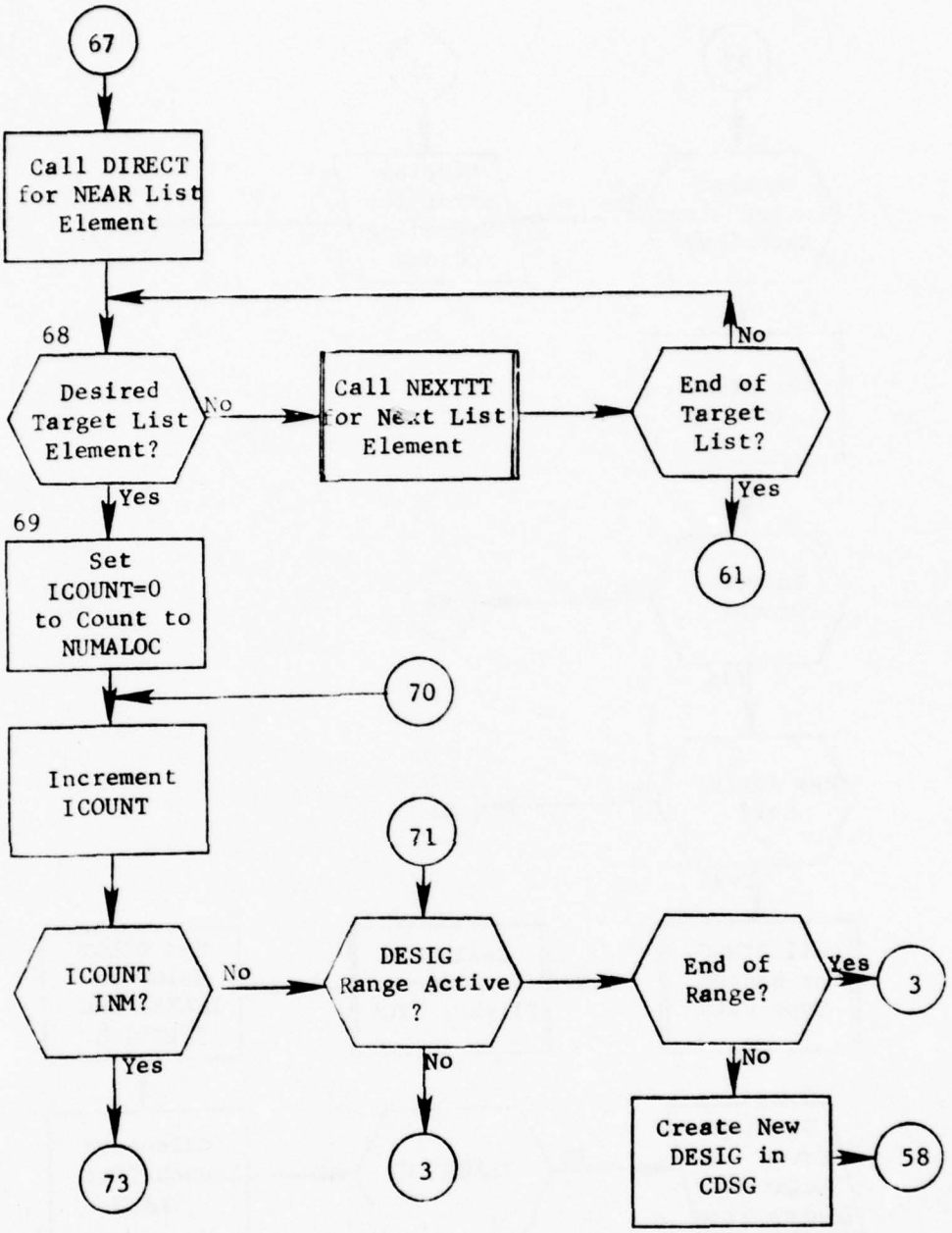


Figure 5. (Part 19 of 23)

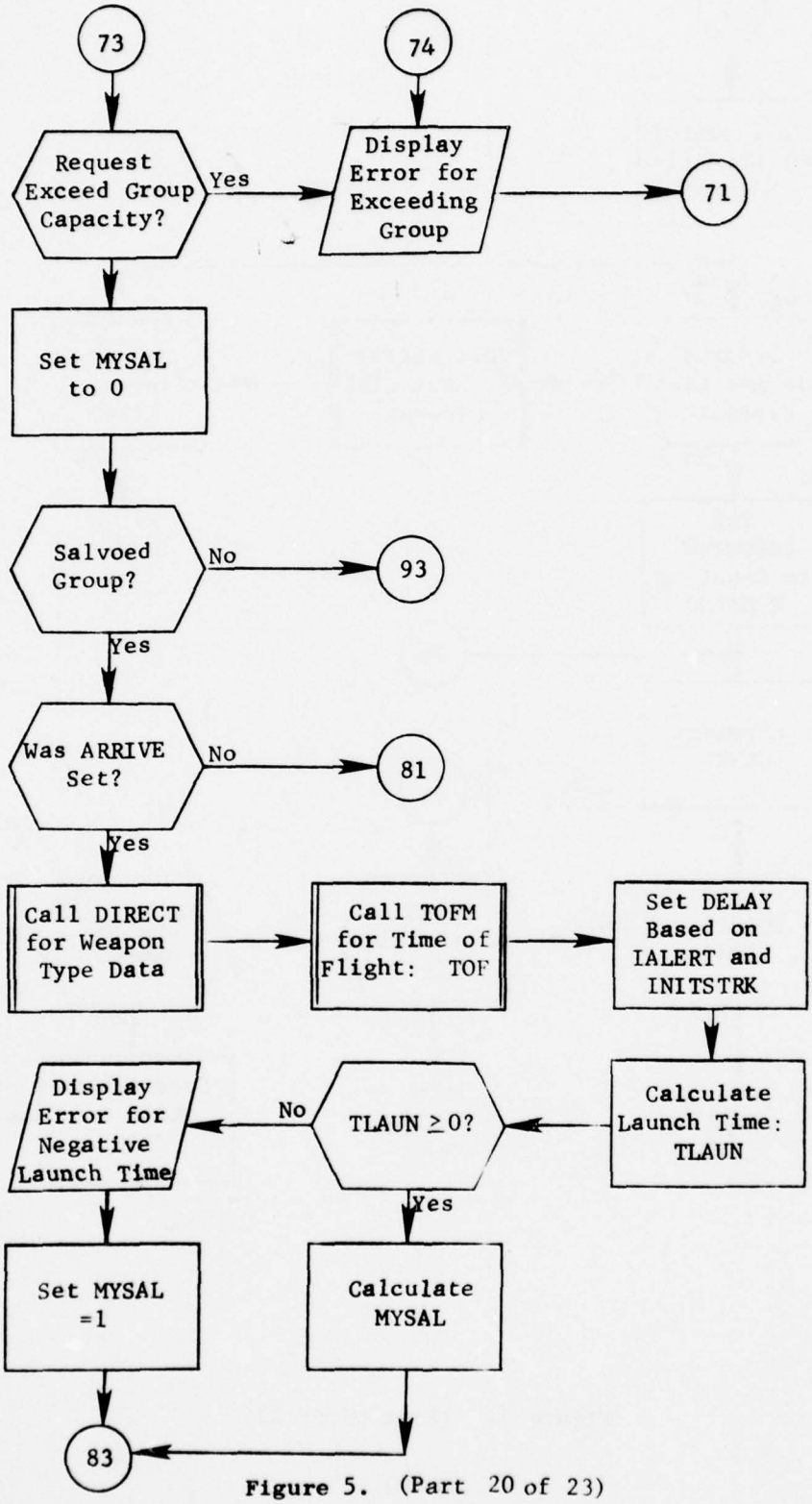


Figure 5. (Part 20 of 23)

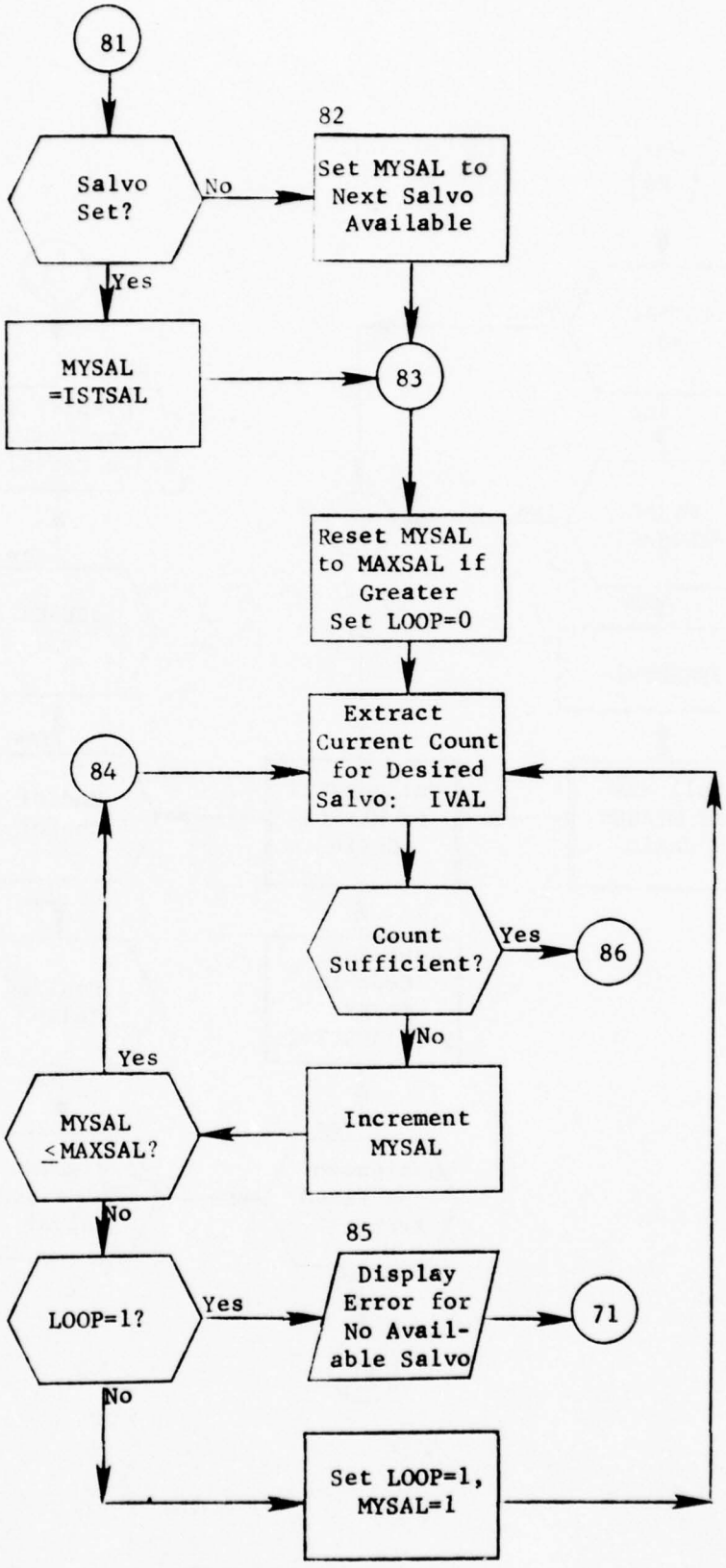


Figure 5. (Part 21 of 23)

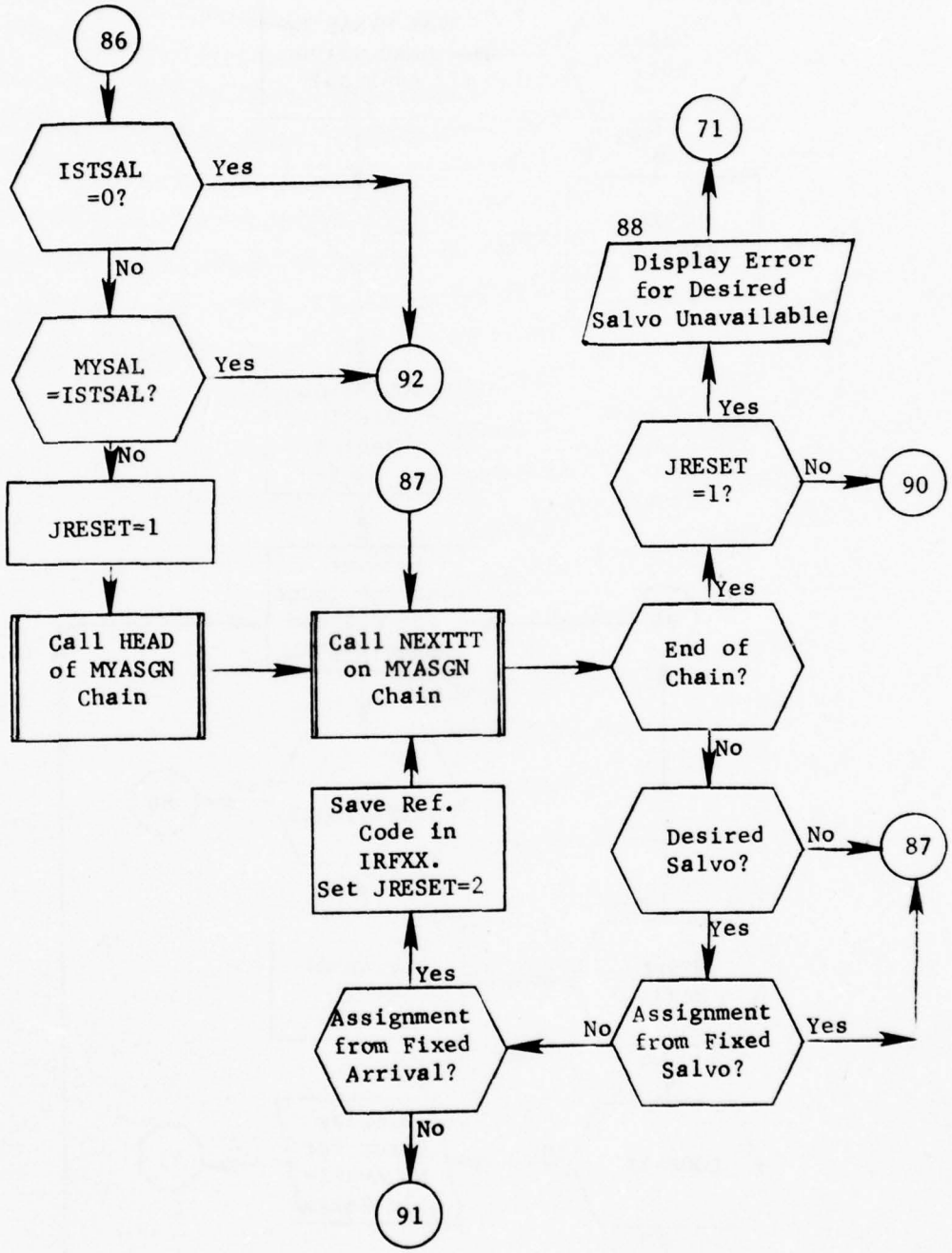


Figure 5. (Part 22 of 23)

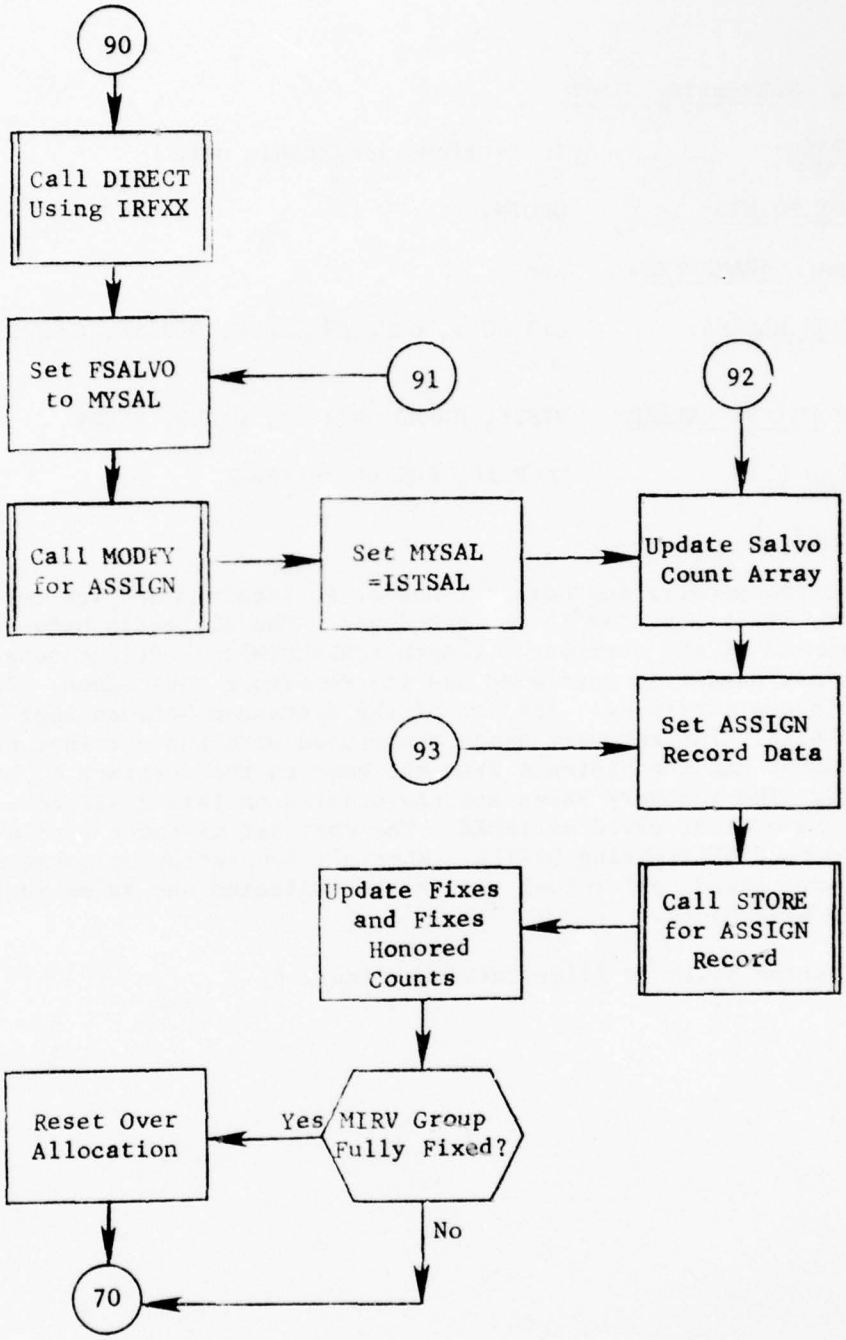


Figure 5. (Part 23 of 23)

2.10 Subroutine GEOIN

PURPOSE: To retrieve geographic data

ENTRY POINTS: GEOIN

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C15, C30, CRLENGTH, DISTEF, GEOREF, RFPOINTS, OOPS

SUBROUTINES CALLED: DISTF, HDFND, NEXTTT, ORDER, RETRV

CALLED BY: GEOPREP, PREPRP, WEPPREP

Method:

First the penetration corridor header is retained and its reference code saved. Each corridor is now retrieved. The distances between all legs is summed as the corridor's length (CRLENGTH). Next the depenetration corridor header is retrieved and its reference code saved. Each corridor is now retrieved. The sum of the distances between legs is stored as DISTEF. The recovery bases associated with the corridor are saved in DRECOV and the distance from the base to the corridor is saved in DISTN. The recovery bases are now ordered on this distance - ascending - and the ordinal saved as INDXA. The shortest distance (ordinal 1) is added to DISTEF giving DISTEG. When all depenetration corridors have been processed, all refuel points are collected and saved in the /RFPOINTS/ block.

Subroutine GEOIN is illustrated in figure 6.

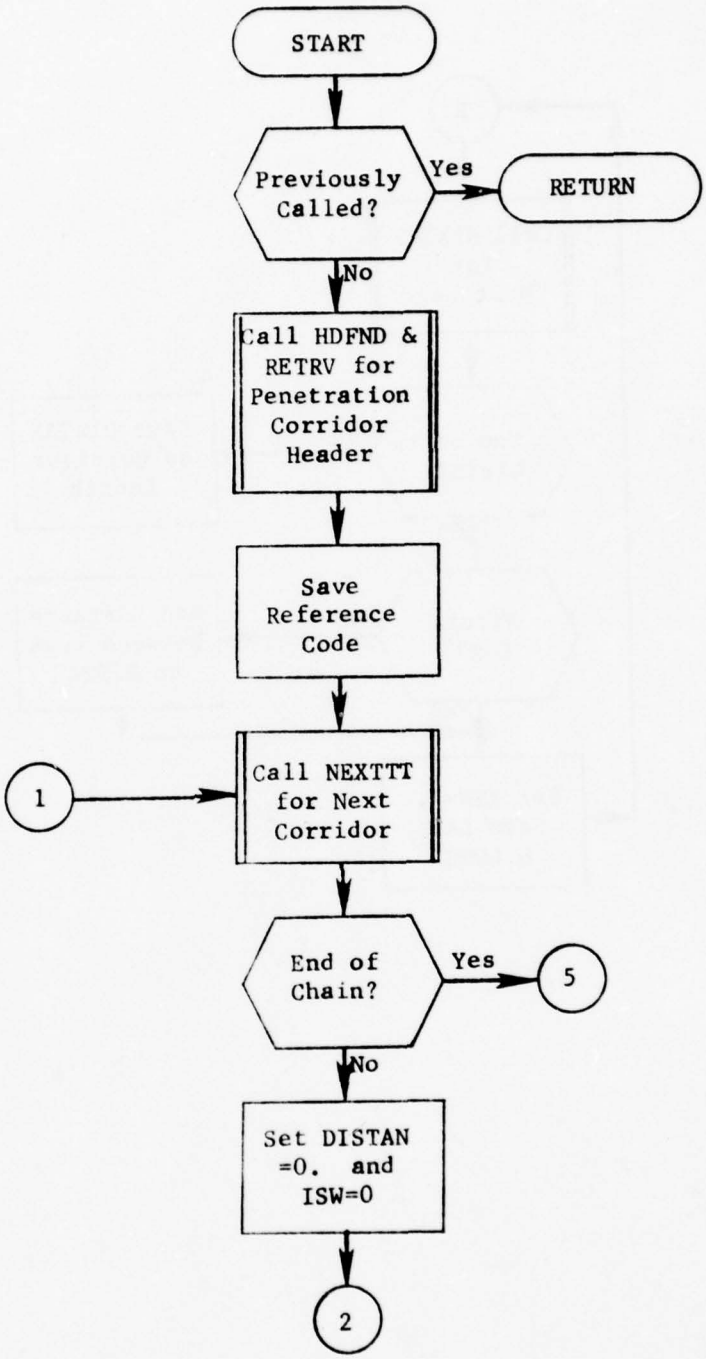


Figure 6. Subroutine GEOIN (Part 1 of 7)

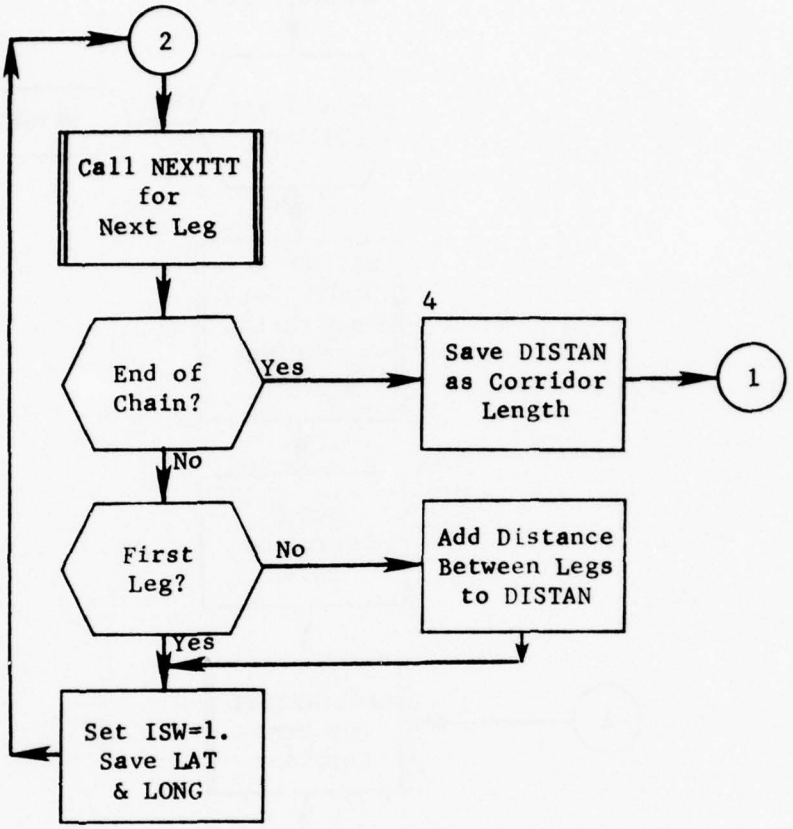


Figure 6. (Part 2 of 7)

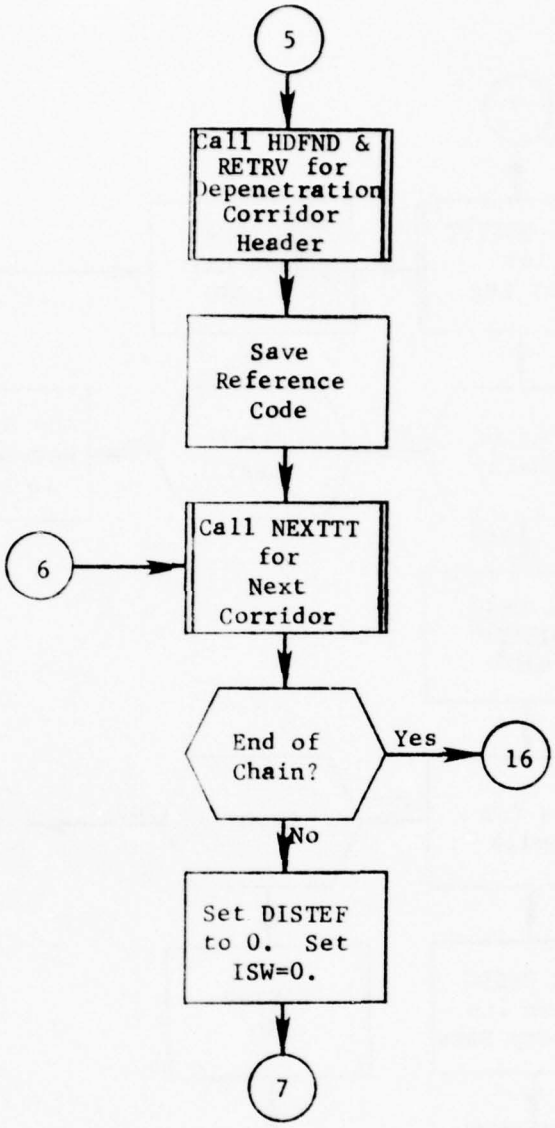


Figure 6. (Part 3 of 7)

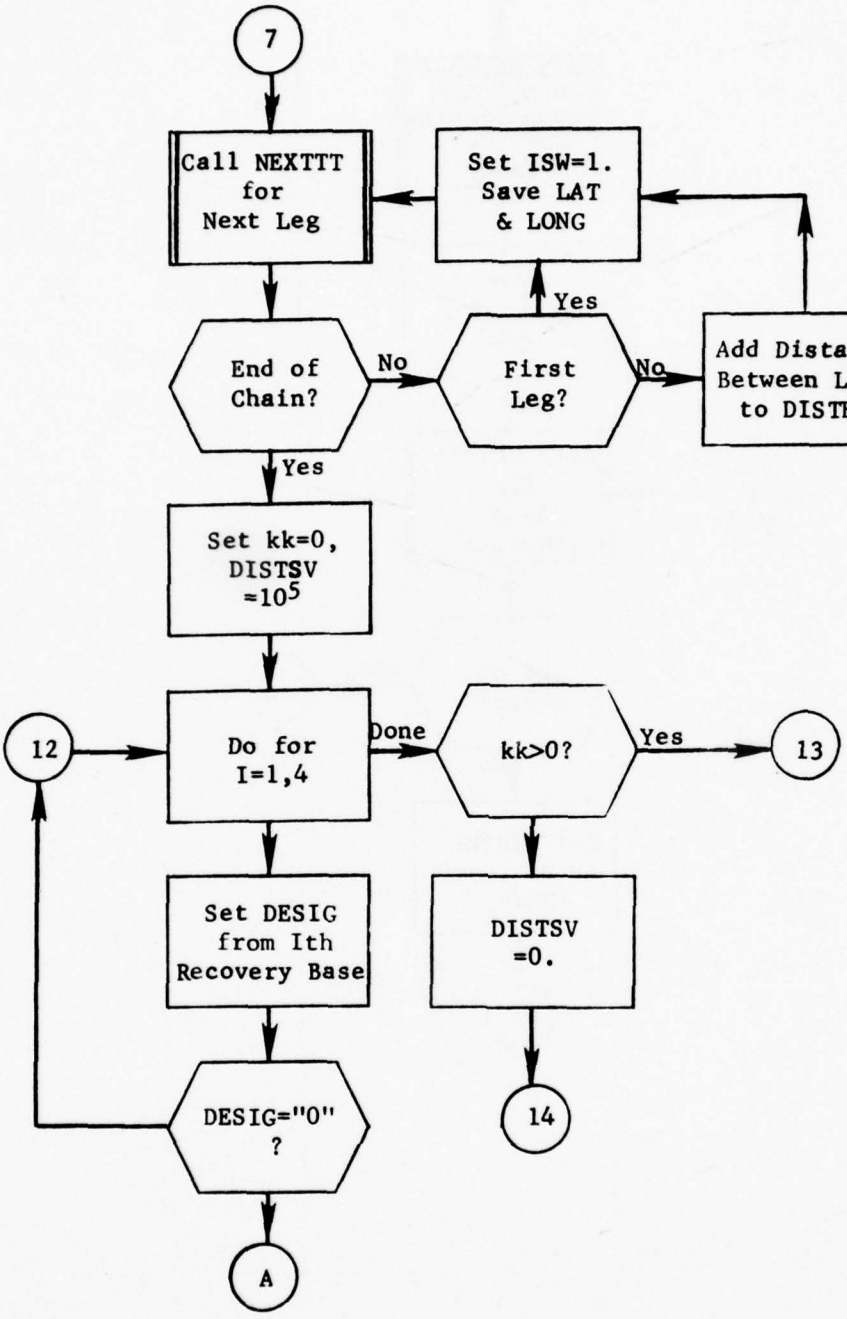


Figure 6. (Part 4 of 7)

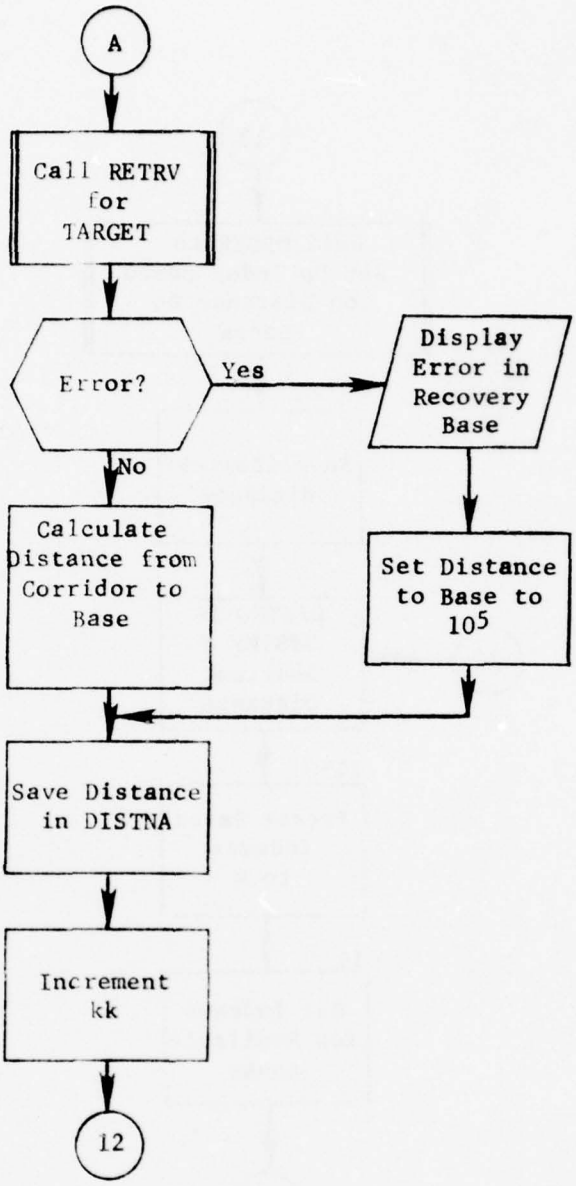


Figure 6. (Part 5 of 7)

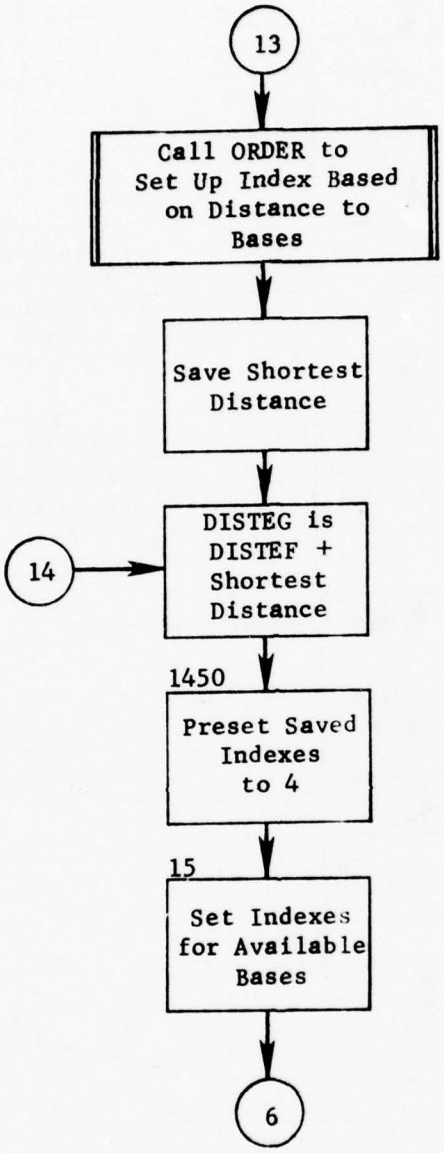


Figure 7. (Part 6 of 7)

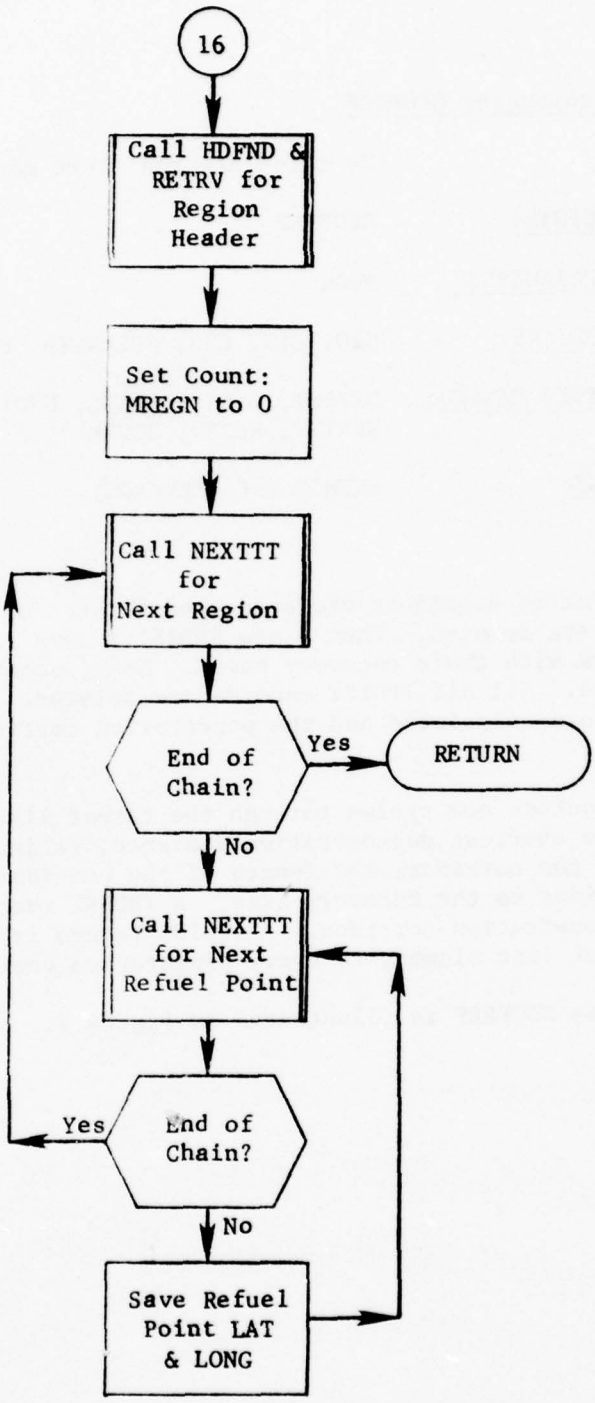


Figure 6. (Part 7 of 7)

2.11 Subroutine GEOPREP

PURPOSE: To calculate and store geographic data

ENTRY POINTS: GEOPREP

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C15, C30, CRLENGTH, DISTEF, GEOREF, OOPS

SUBROUTINES CALLED: DIRECT, DISTF, DLETE, GEOIN, HDFND, HEAD, MODFY, NEXTTT, RETRV, STORE

CALLED BY: ENTMOD (of PREPALOC)

Method:

Depenetration corridors are processed first. All old TDDIST and RDDIST records are deleted. Then a new RDDIST record is stored connecting the corridors with their recovery bases. Next, penetration corridors are processed. All old TPDIST records are deleted. Then all attributes sectors are calculated and the penetration corridor record (PENCRD) modified.

The subroutine now cycles through the target list. For each target it finds the shortest depenetration distance, taking into account the distance to the corridor, the length of the corridor and the distance from the corridor to the recovery base. A TDDIST record is created for the best depenetration corridor. A TPDIST record is also stored correcting the target list element to every penetration corridor.

Subroutine GEOPREP is illustrated in figure 7.

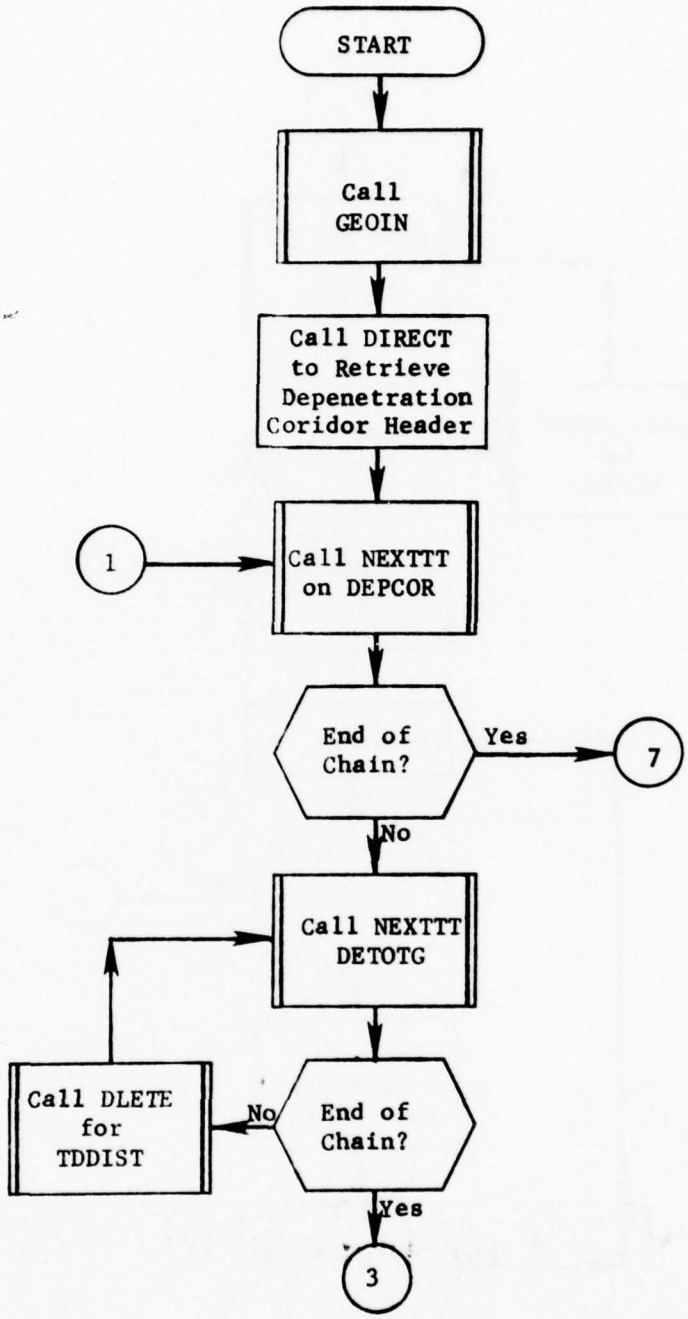


Figure 7. Subroutine GEOPREP (Part 1 of 8)

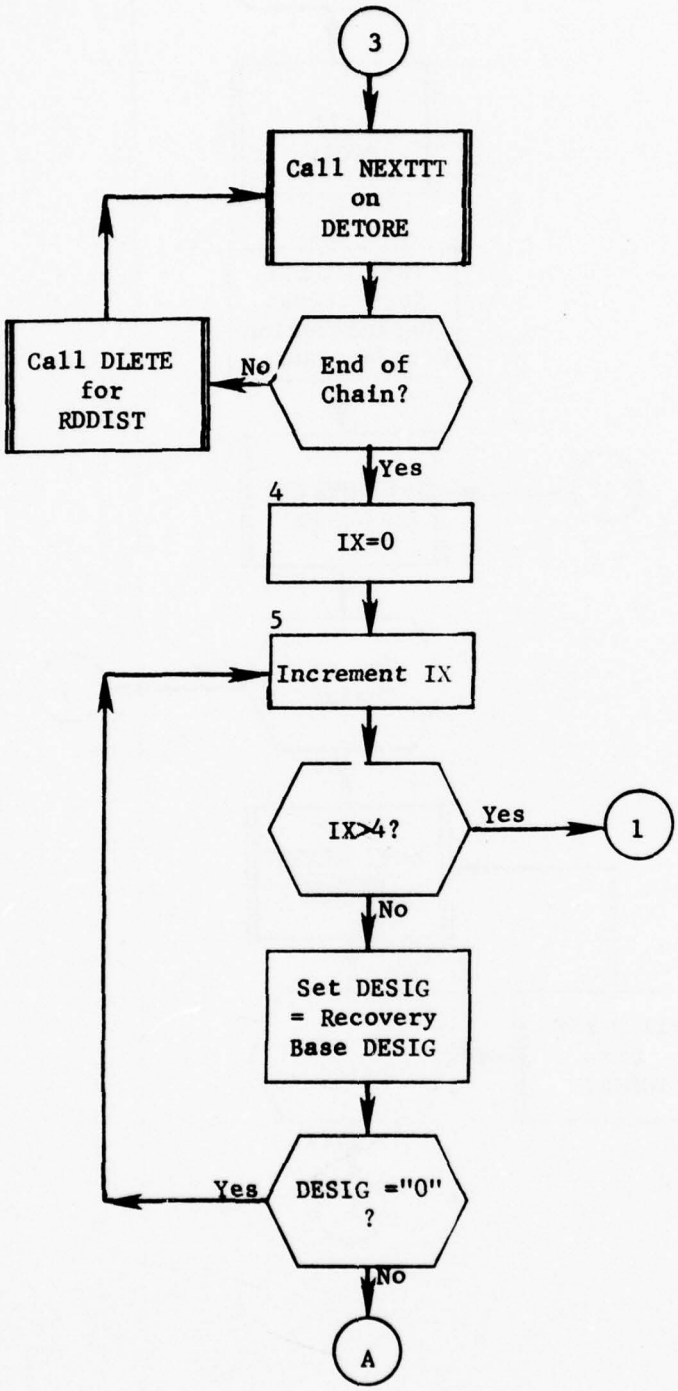


Figure 7. (Part 2 of 8)

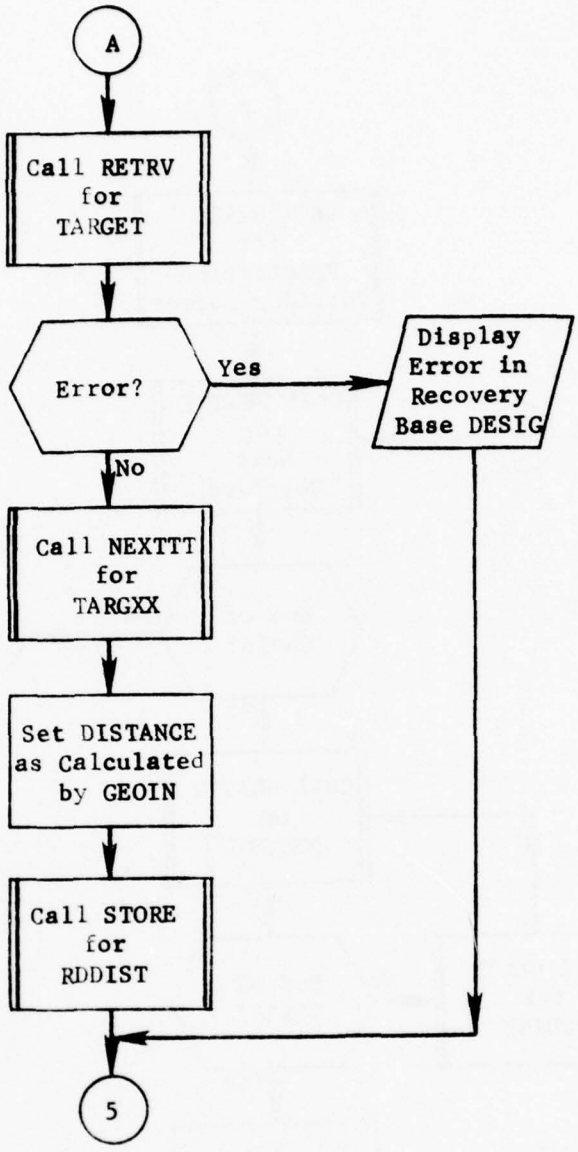


Figure 7. (Part 3 of 8)

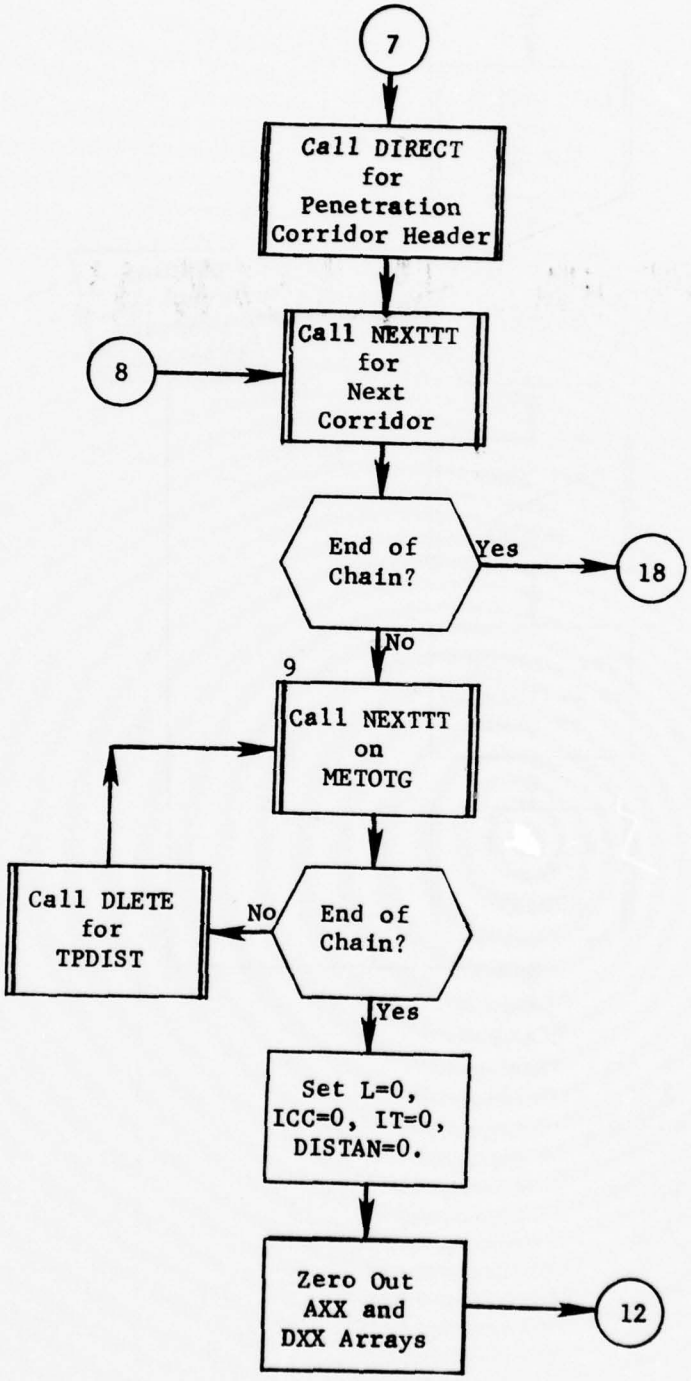


Figure 7. (Part 4 of 8)

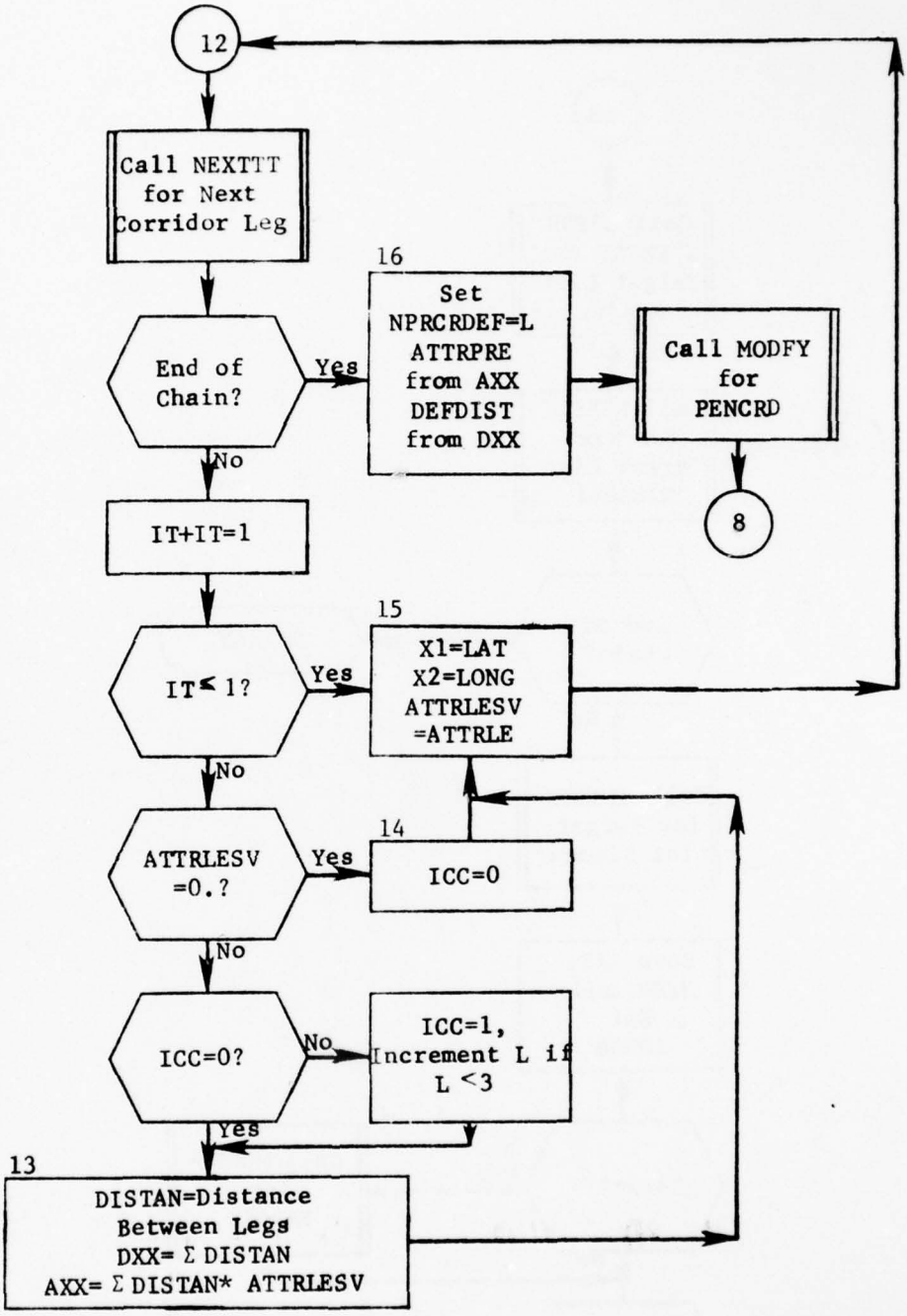


Figure 7. (Part 5 of 8)

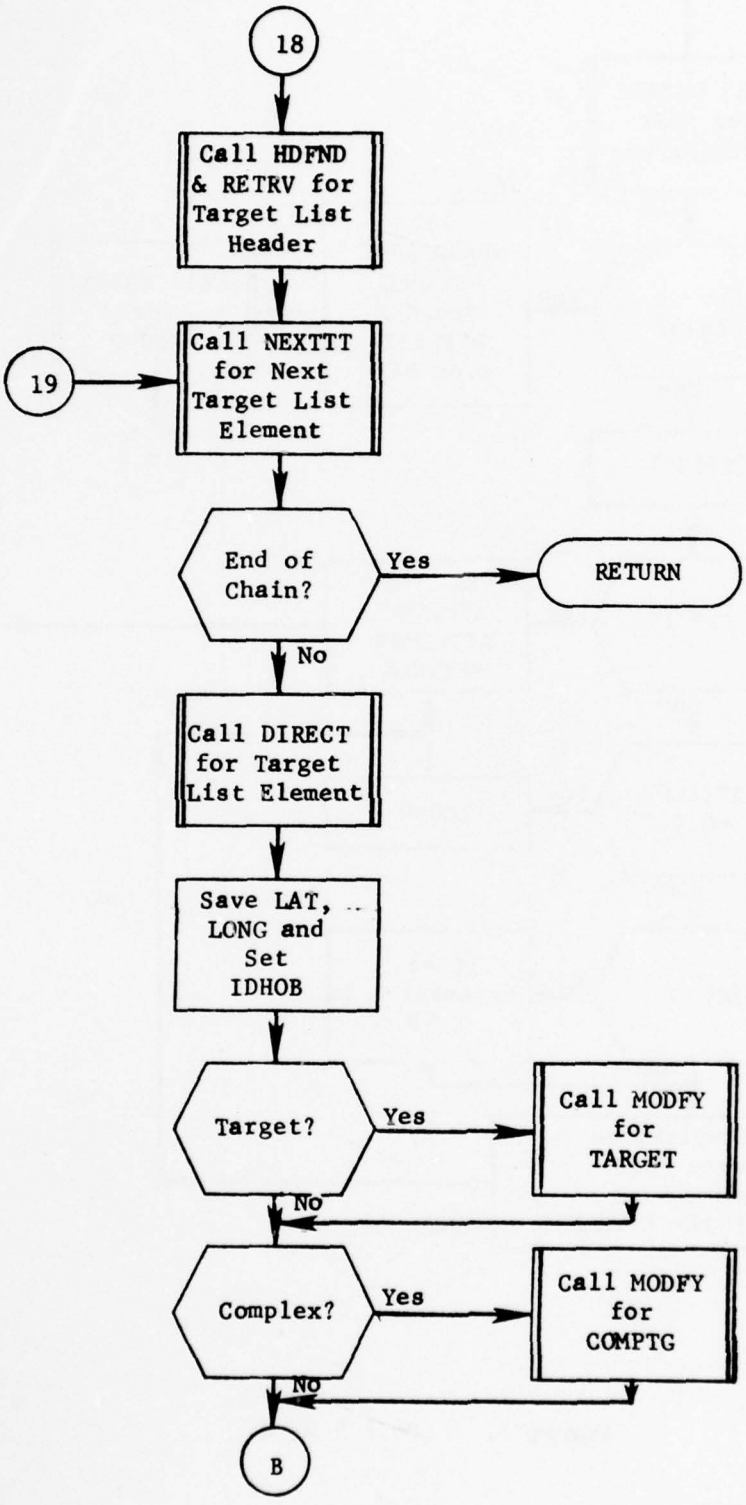


Figure 7. (Part 6 of 8)

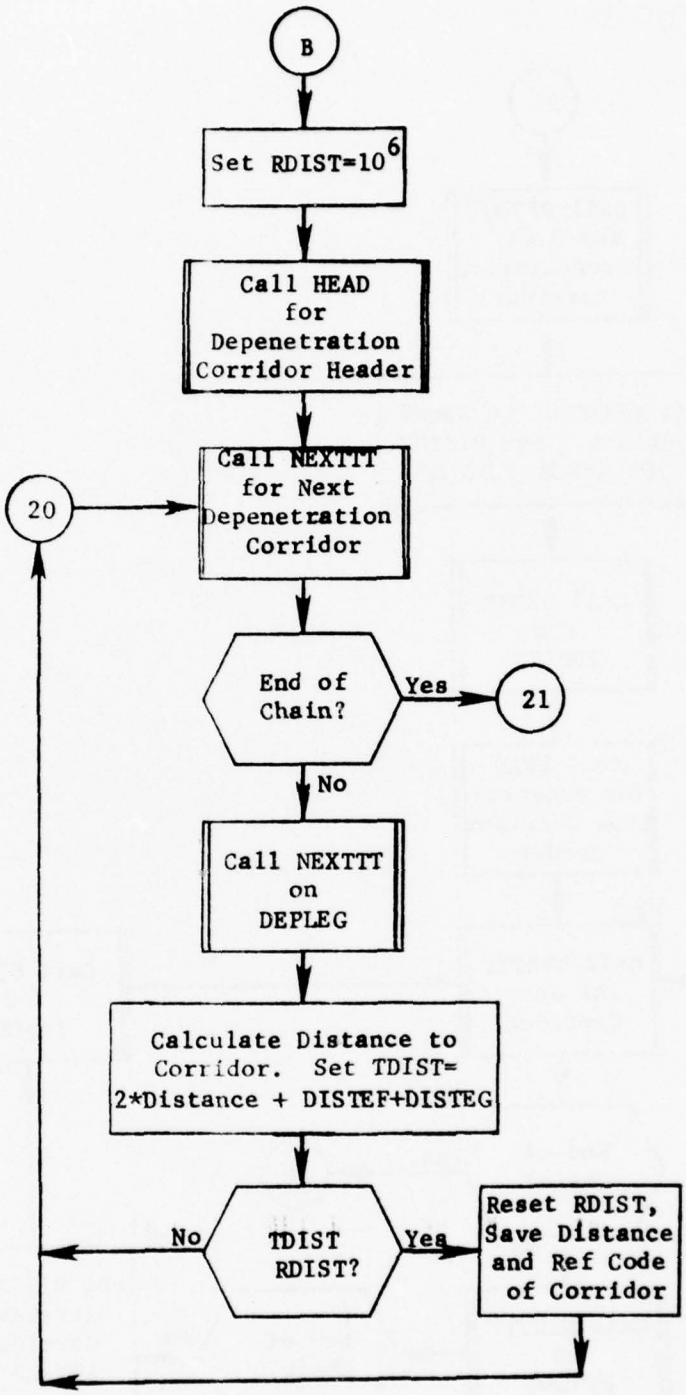


Figure 7. (Part 7 of 8)

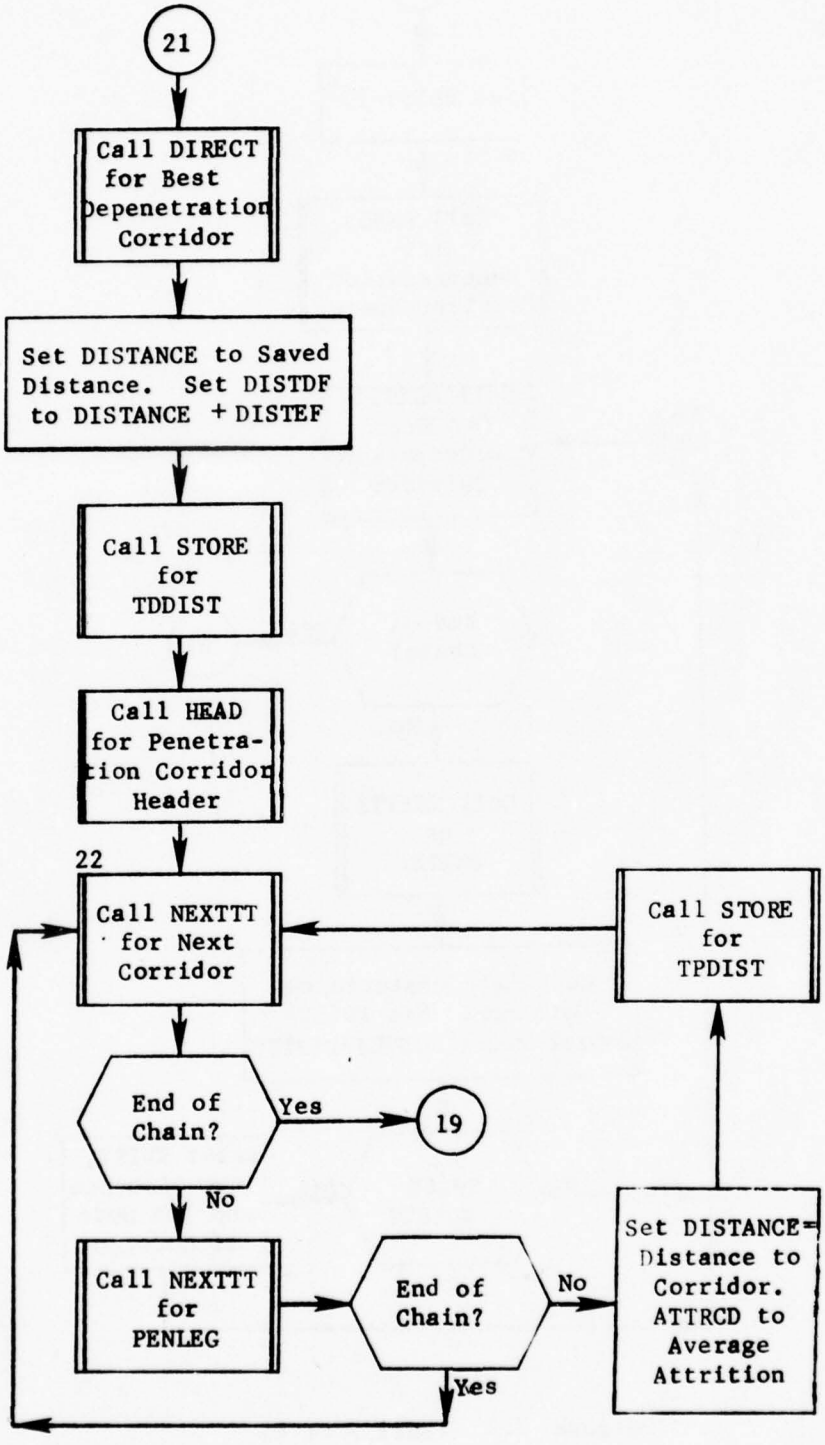


Figure 7. (Part 8 of 8)

2.12 Subroutine PRNPRP

PURPOSE: To perform all PREPALOC prints

ENTRY POINTS: PRNPRP

FORMAL PARAMETERS: ICPT - index of ONPRINTS clause

COMMON BLOCKS: C10, C15, C30, CRLENGTH, DISTEF, GEOREF, IFXREQ, IFXSW, NTBFLG, OOPS, RFPOINTS, SETCOM, WEPCOM, ZEES

SUBROUTINES CALLED: DIRECT, DISTF, GEOIN, HDFND, HEAD, INSGET, NEXTTT, RETRV, WEPIN

CALLED BY: ENTMOD (of PREPALOC)

Method:

The operation of this subroutine depends largely on the ONPRINTS clause and the options the user has exercised. The ONPRINTS clause (if any) is read and the selected options saved as switches. If the RECALC option is active or there have been changes to the general gaming parameters, the "USER INPUT PLANNING PARAMETERS" print is produced. If any FIX clauses were present in the input the "FIXED ASSIGNMENTS" print is produced. If the user has selected print option 1 the geographic data is printed. If the user has selected print option 2 the distances of groups to corridors are printed. If the user has selected print option 3, the "TARGET INFORMATION PRINT" is produced. If the user has selected print option 4, the "PLANNING FACTORS print is produced. Finally, if the RECALC mode is active or there have been changes to the target data base, the class value summary is printed.

Subroutine PRNPRP is illustrated in figure 8.

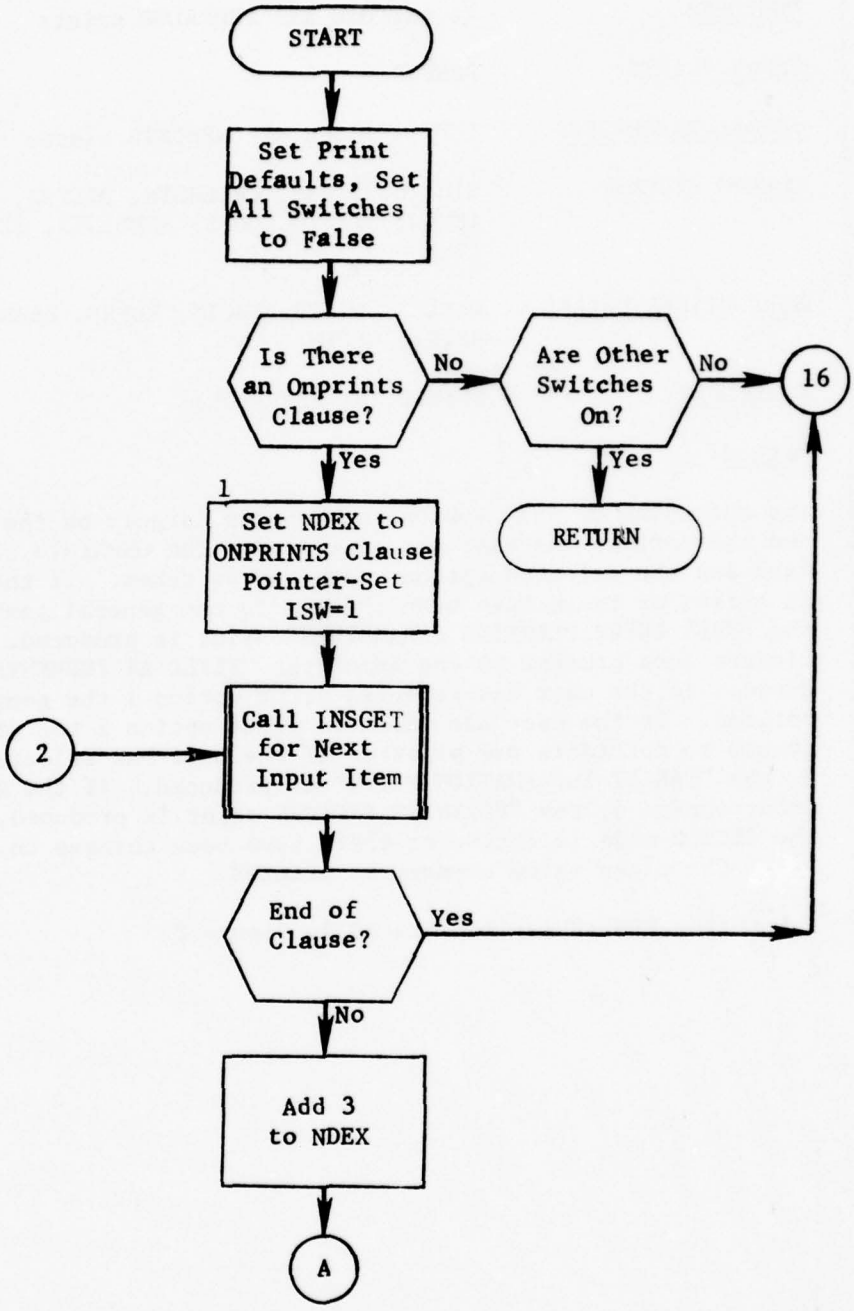


Figure 8. Subroutine PRNPRP (Part 1 of 12)

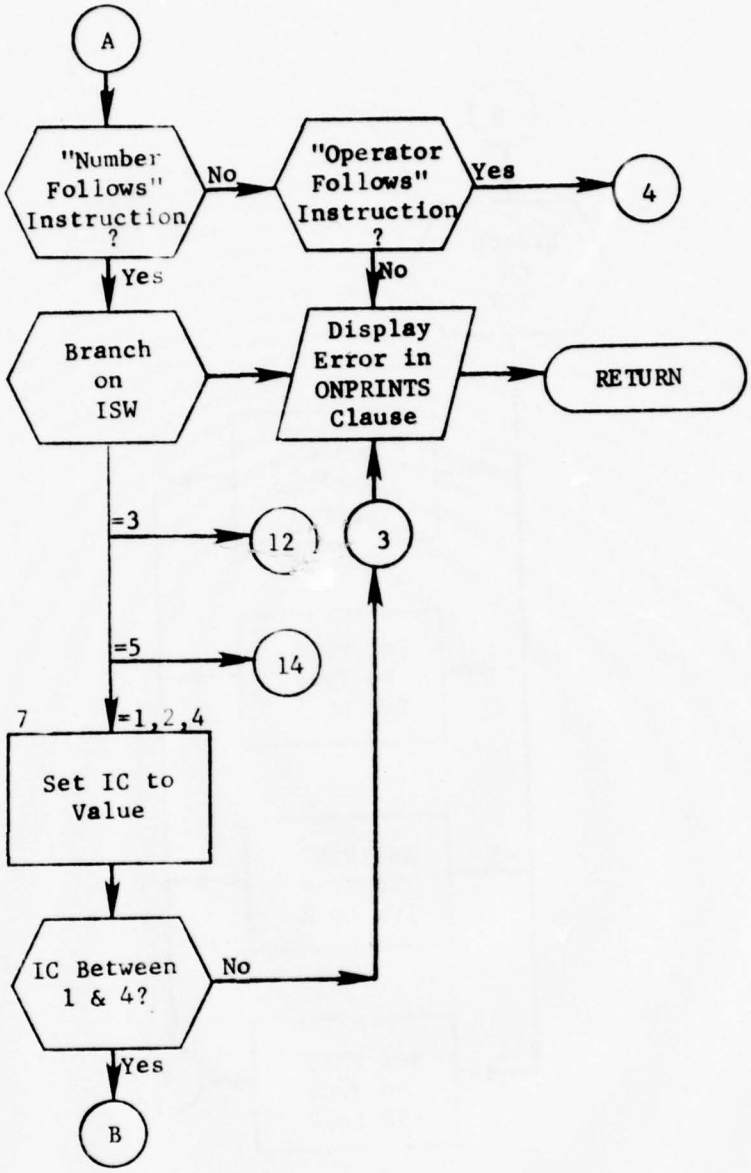


Figure 8. (Part 2 of 12)

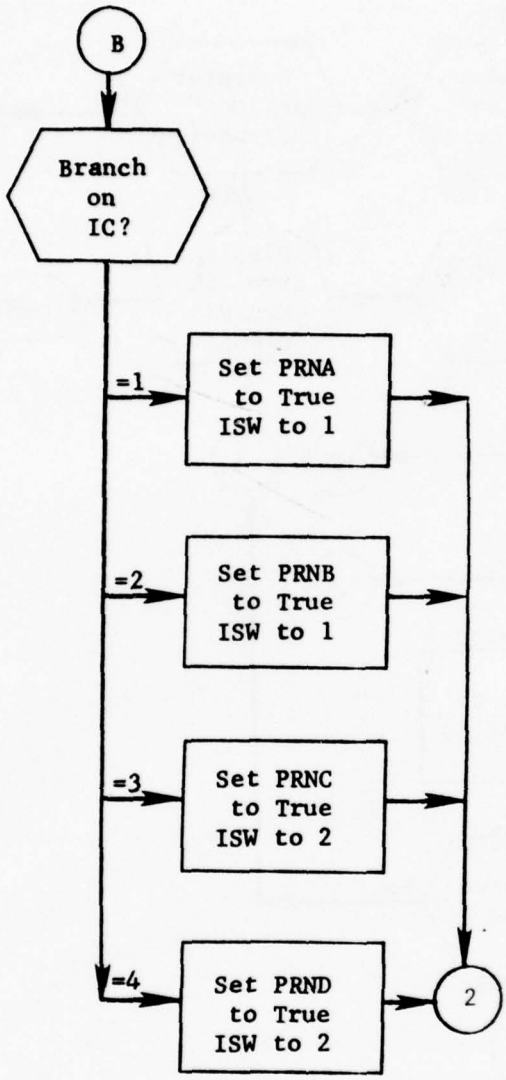


Figure 8. (Part 3 of 12)

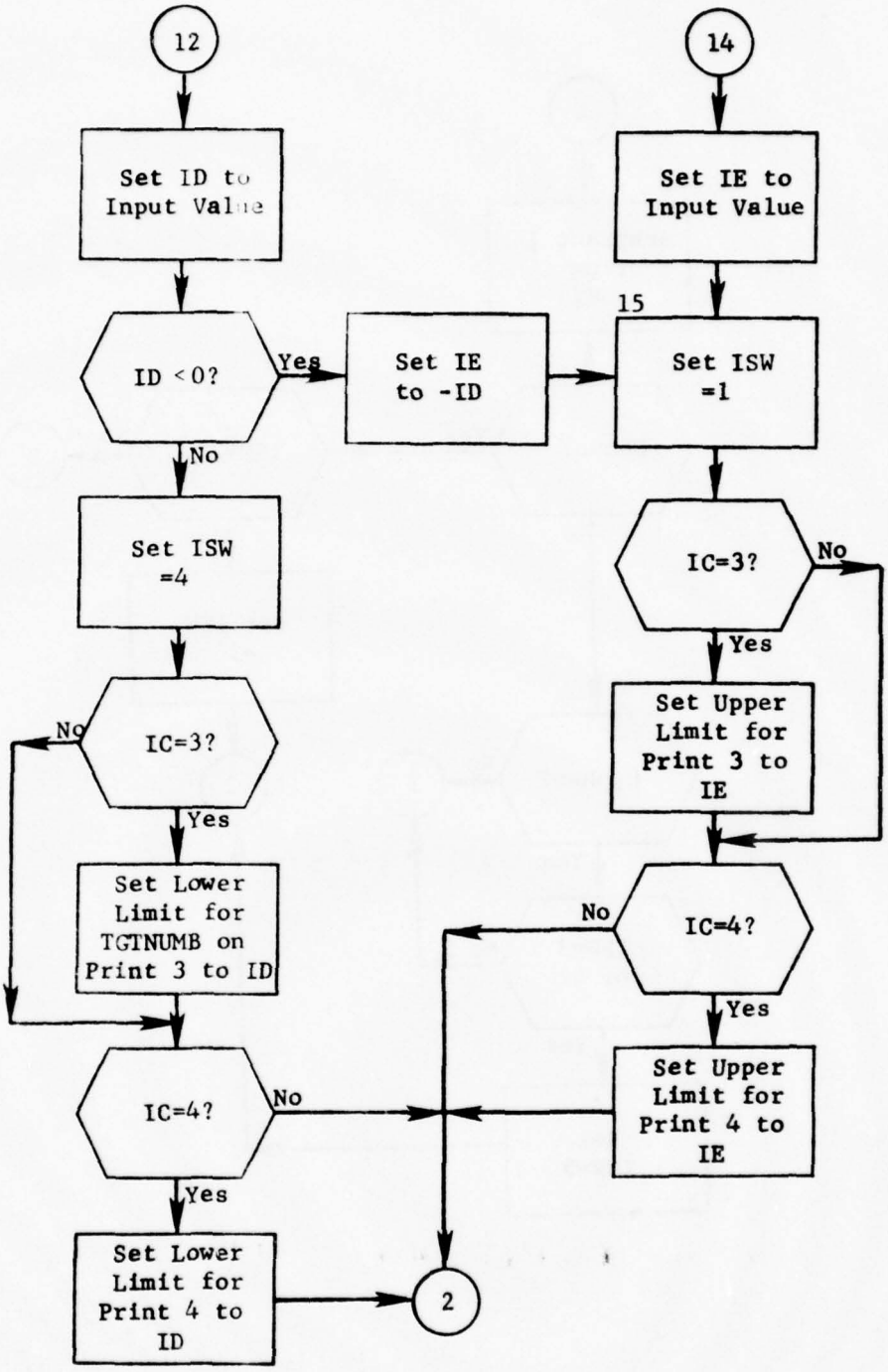


Figure 8. (Part 4 of 12)

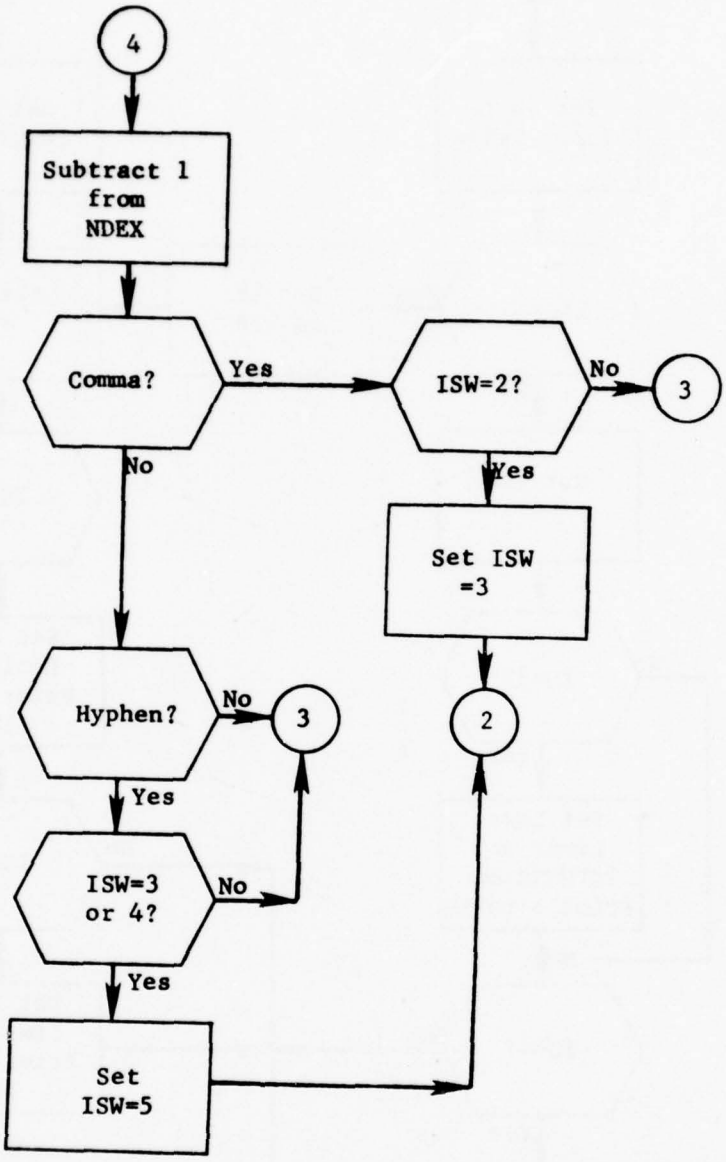


Figure 8. (Part 5 of 12)

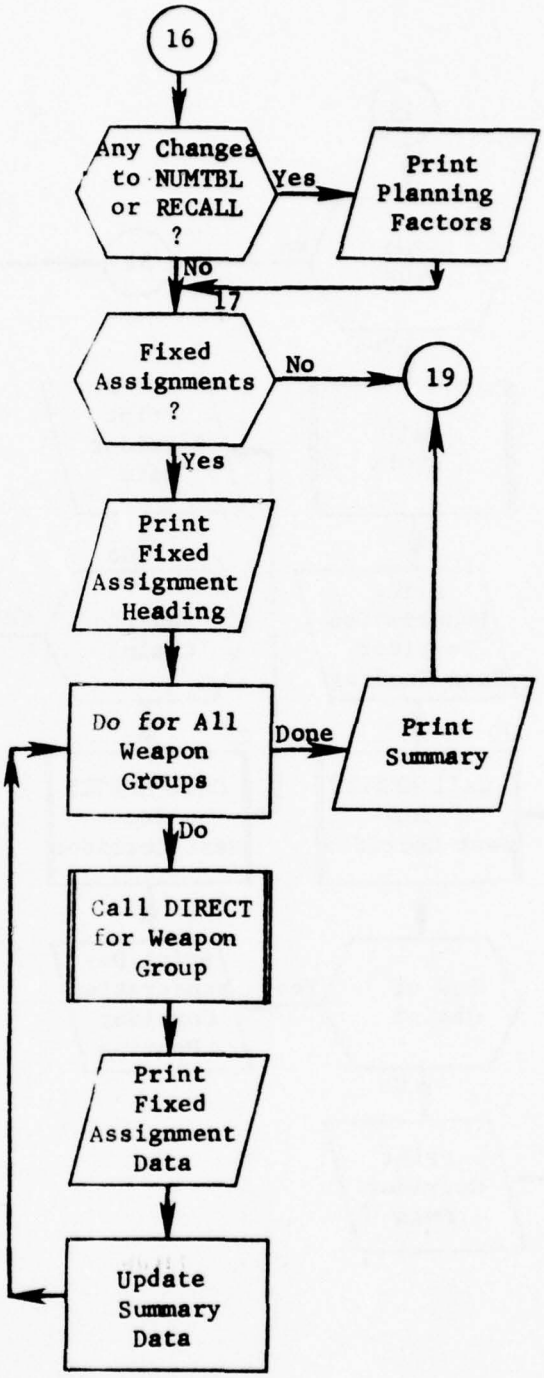


Figure 8. (Part 6 of 12)

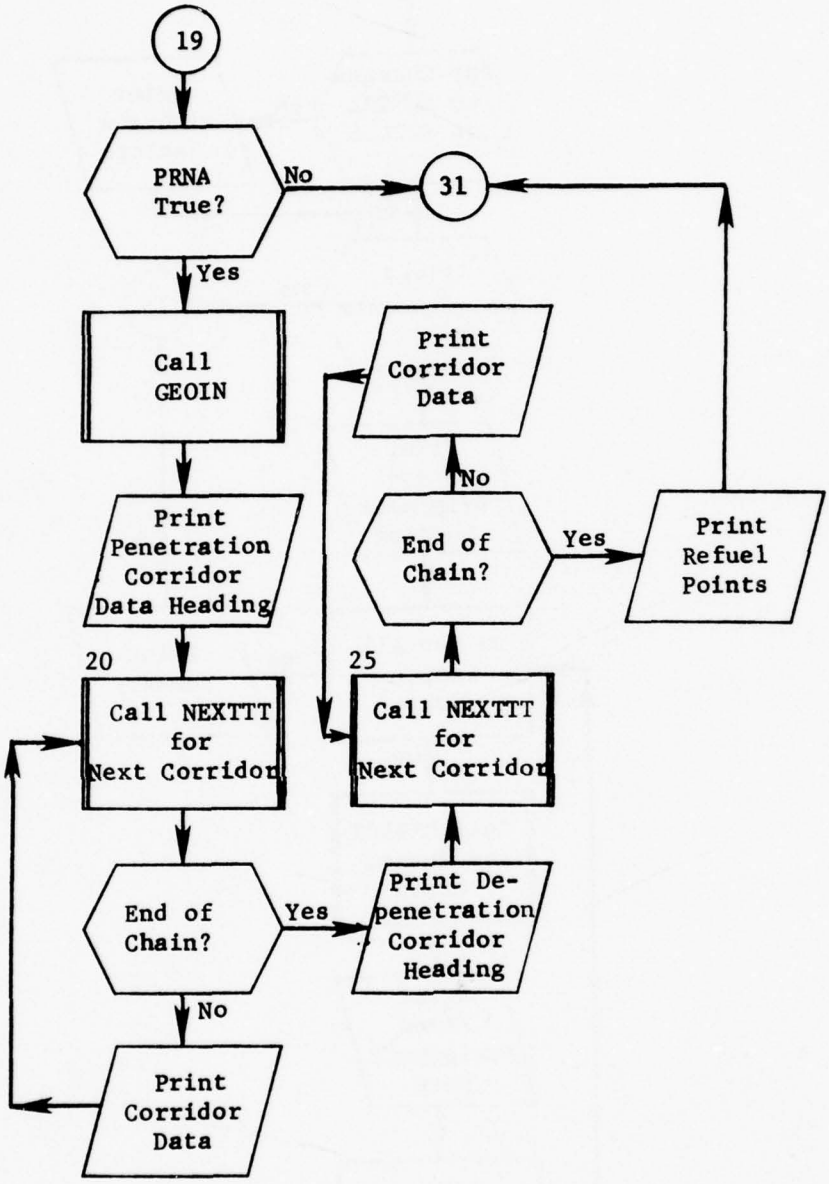


Figure 8. (Part 7 of 12)

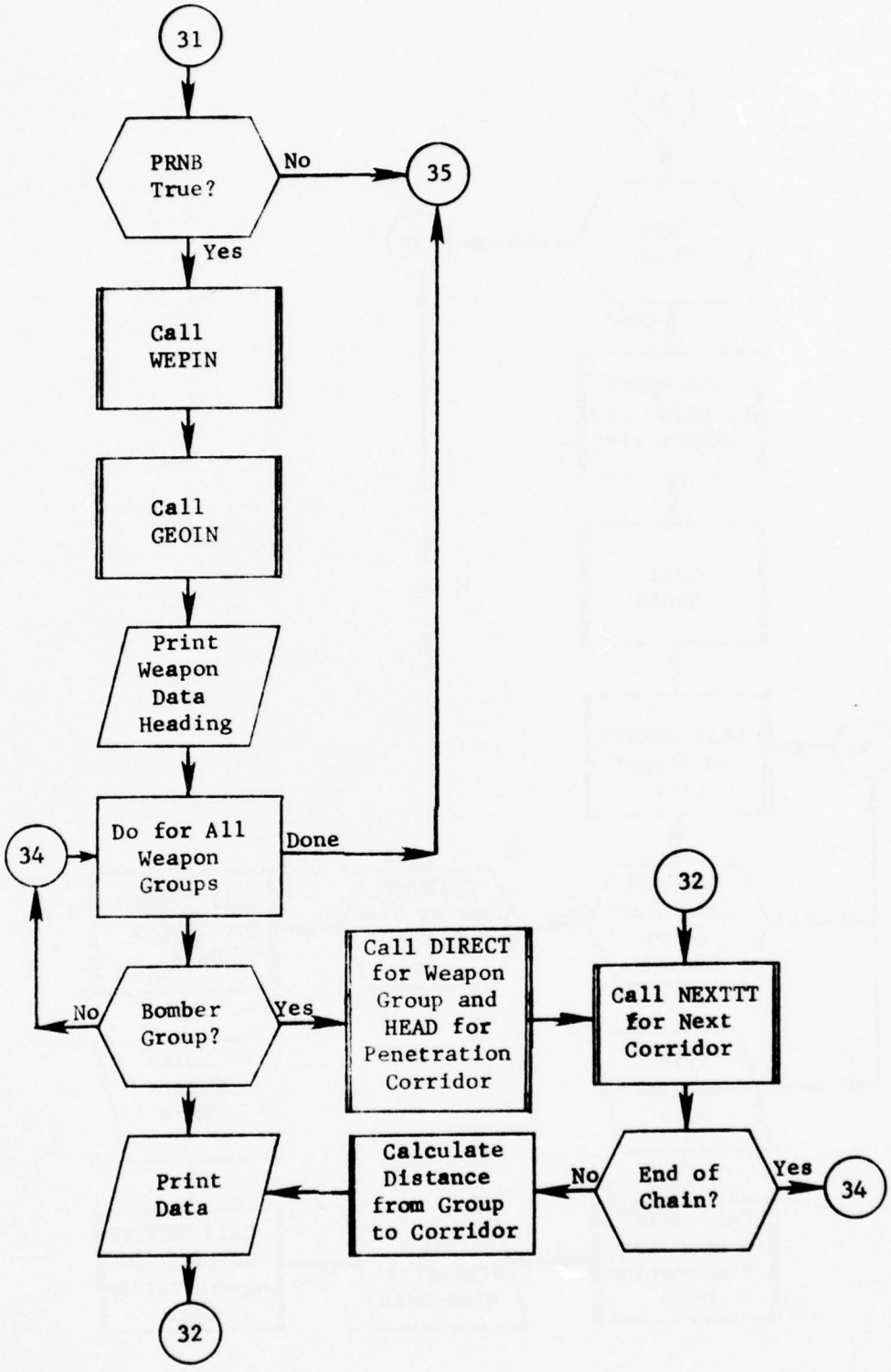


Figure 8. (Part 8 of 12)

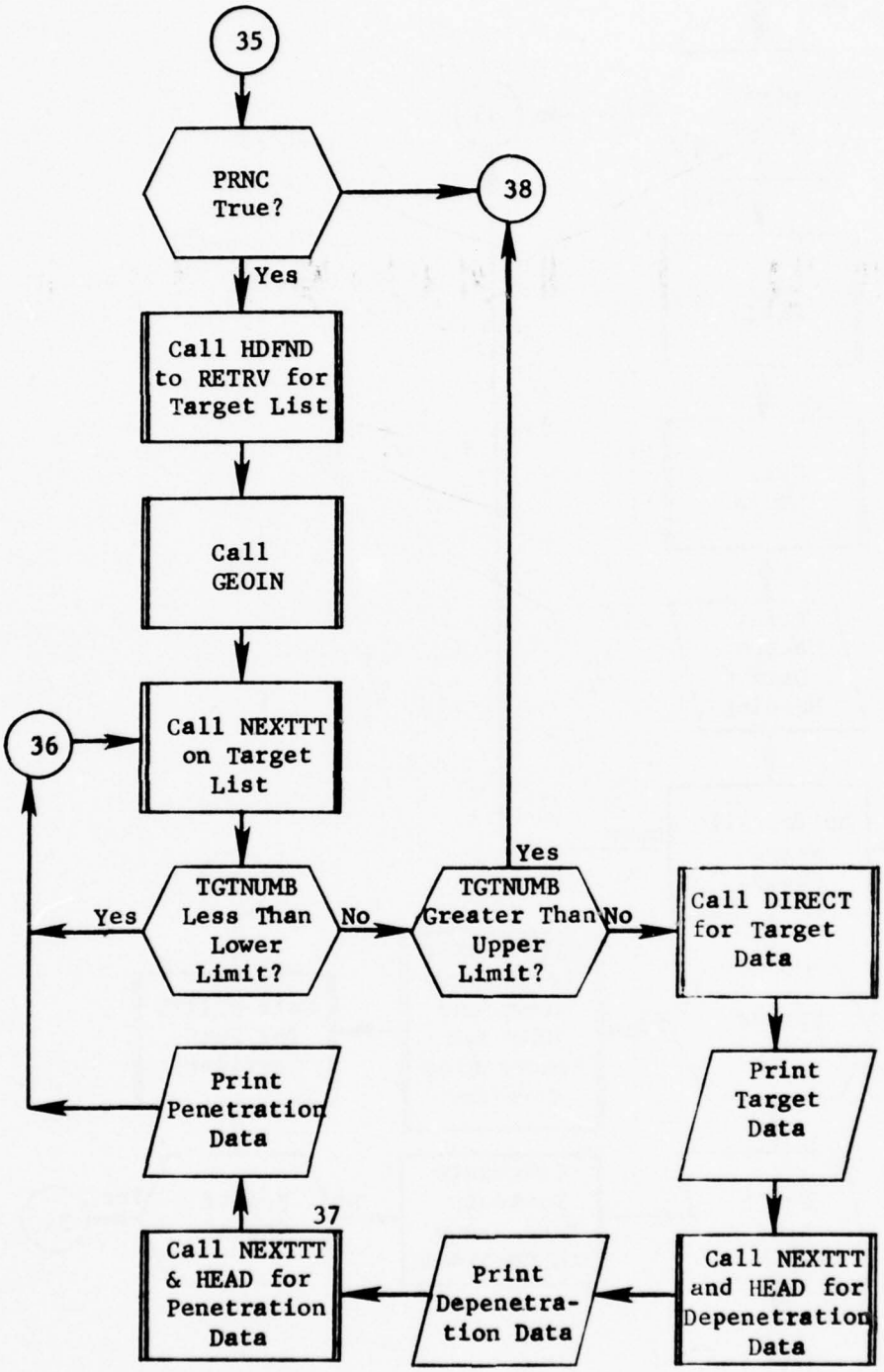


Figure 8. (Part 9 of 12)

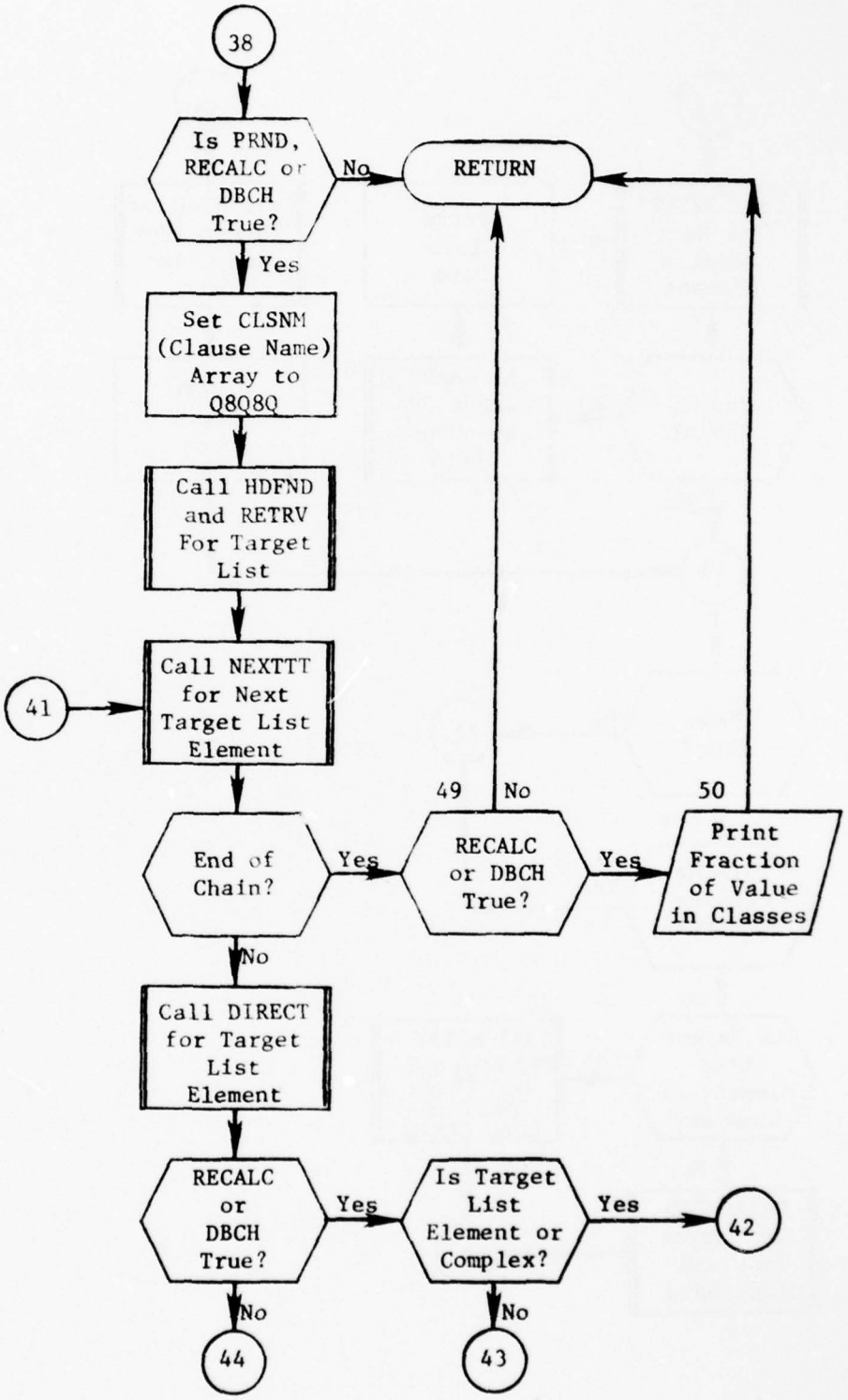


Figure 8. (Part 10 of 12)

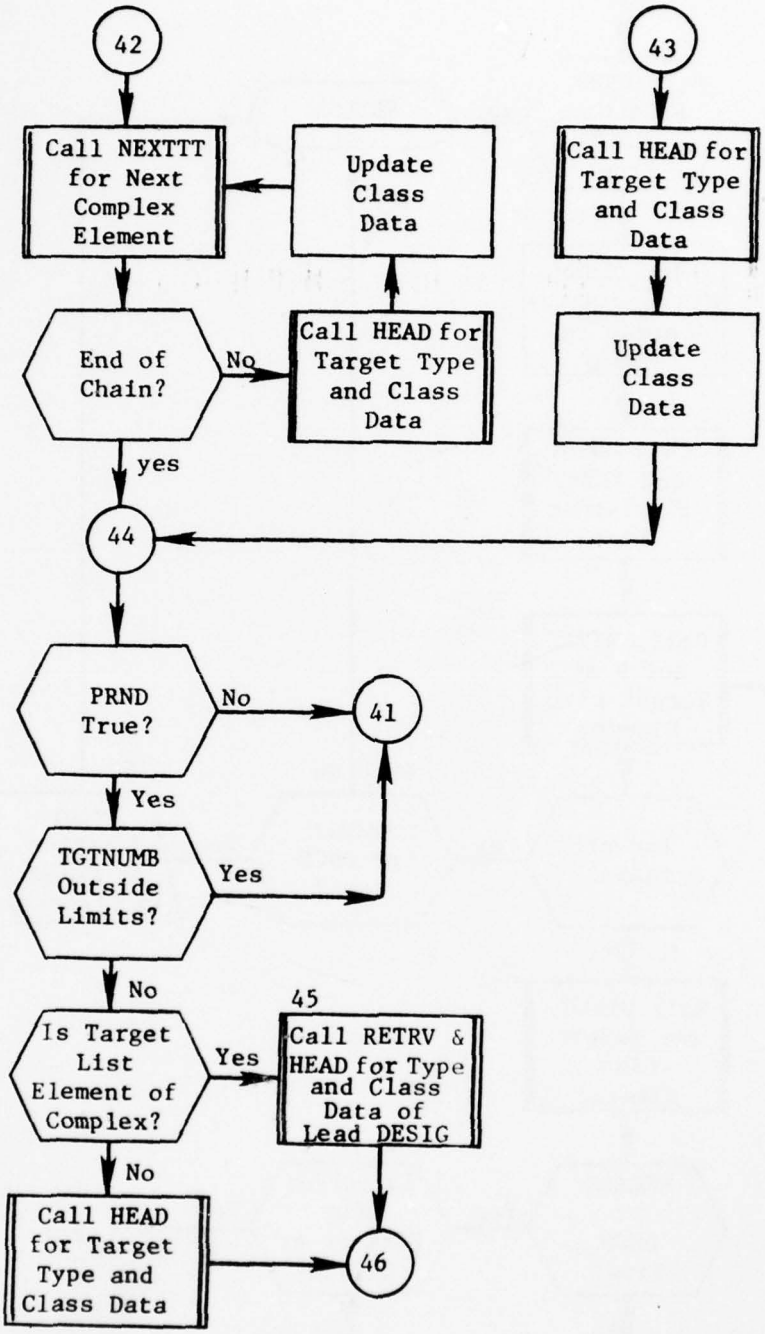


Figure 8. (Part 11 of 12)

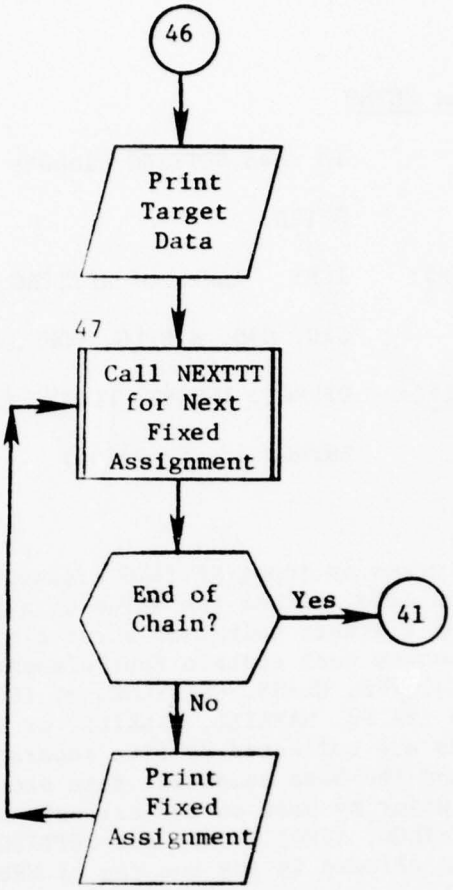


Figure 8. (Part 12 of 12)

2.13 Subroutine SETRD

PURPOSE: To read SETTING clauses

ENTRY POINTS: SETRD

FORMAL PARAMETERS: ICPT - index of SETTING clause

COMMON BLOCK: C10, C30, NTBFLG, OOPS, SETCOM, WEPCOM

SUBROUTINES CALLED: DIRECT, INSGET, MODFY, RANSIZ, WEPIN

CALLED BY: ENTMOD (of PREPALOC)

Method:

This subroutine scans an input SETTING clause for data base changes. Any changes which alter either the value of a general gaming parameter or a weapon group's preset height of burst are immediately carried out. The remaining changes each contain four elements: a change criteria attribute (DESIG, TYPE, CLASS, CNTRYLOC, or IREG), a criteria value, a change attribute (VALUE, MAXKILL, MINKILL, or IDHOB) and a change value. These change sets are collected by five separate passes through the SETTING clause and the base essential data stored on file 25. The passes are in a priority base on the criteria attribute (order in DESIG, TYPE, CLASS, CNTRYLOC, IREG). Only one SETTING clause containing this type of change is allowed in any one run of PREPALOC

Subroutine SETRD is illustrated in figure 9.

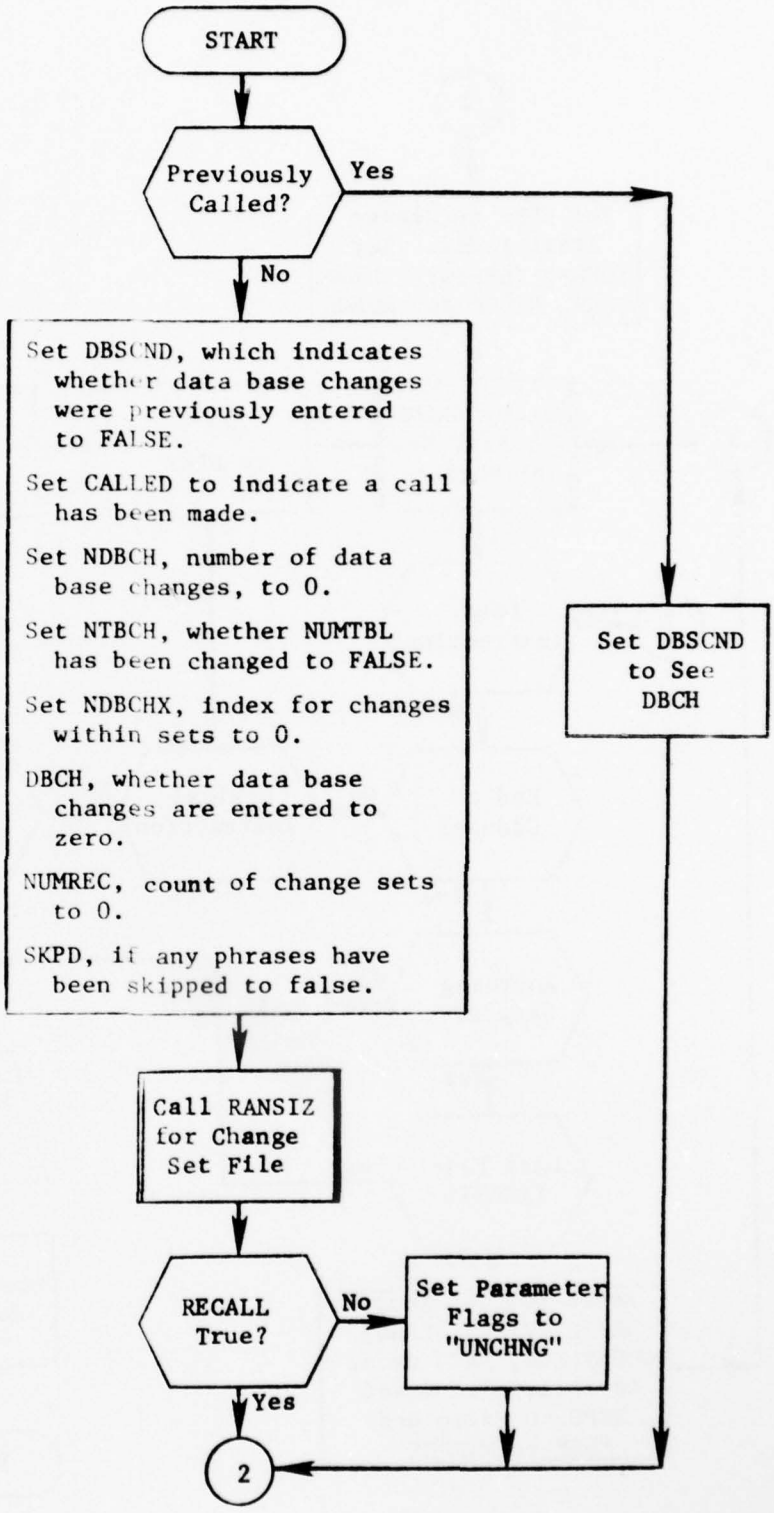


Figure 9. Subroutine SETRD (Part 1 of 5)

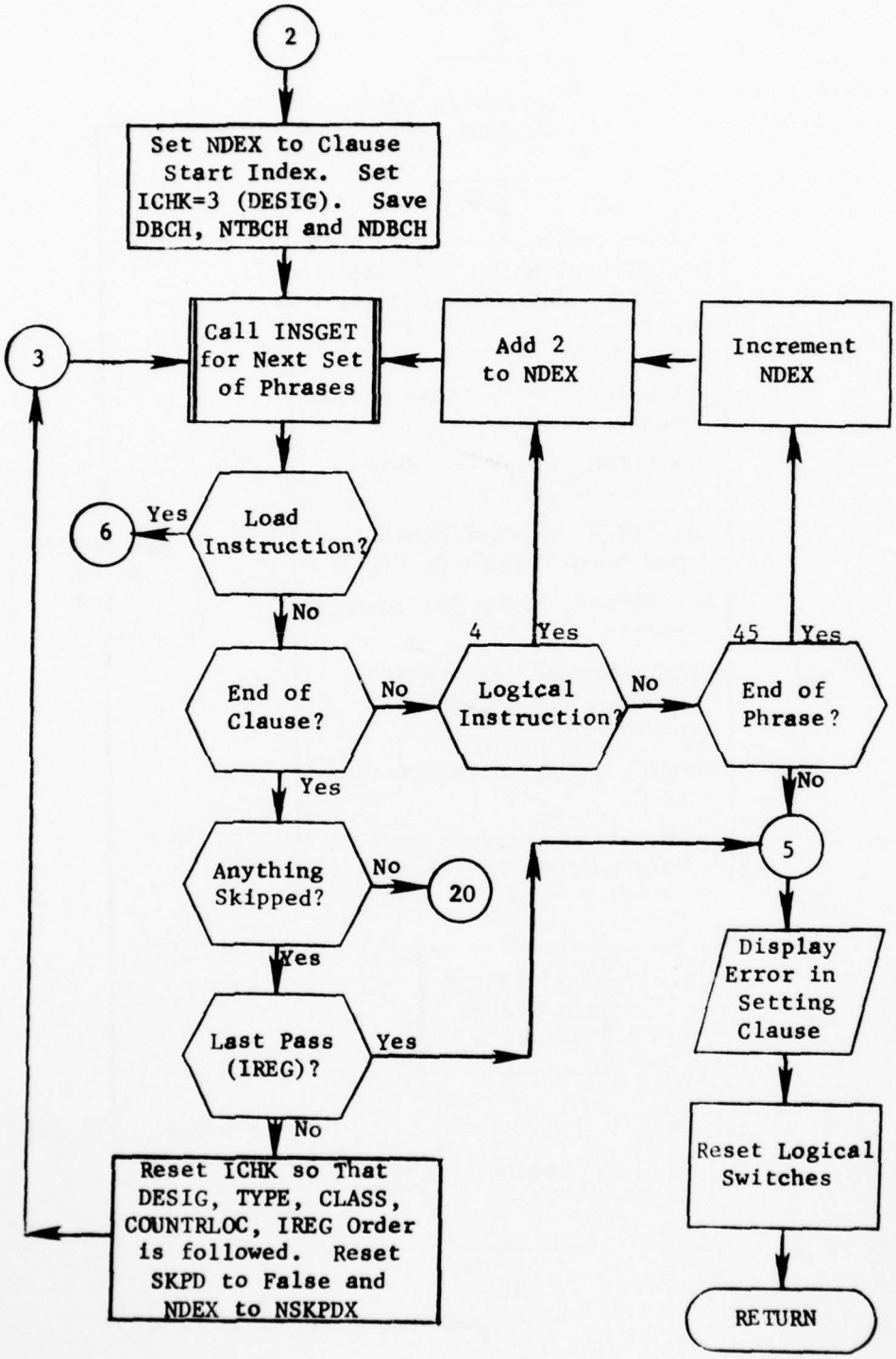


Figure 9. (Part 2 of 5)

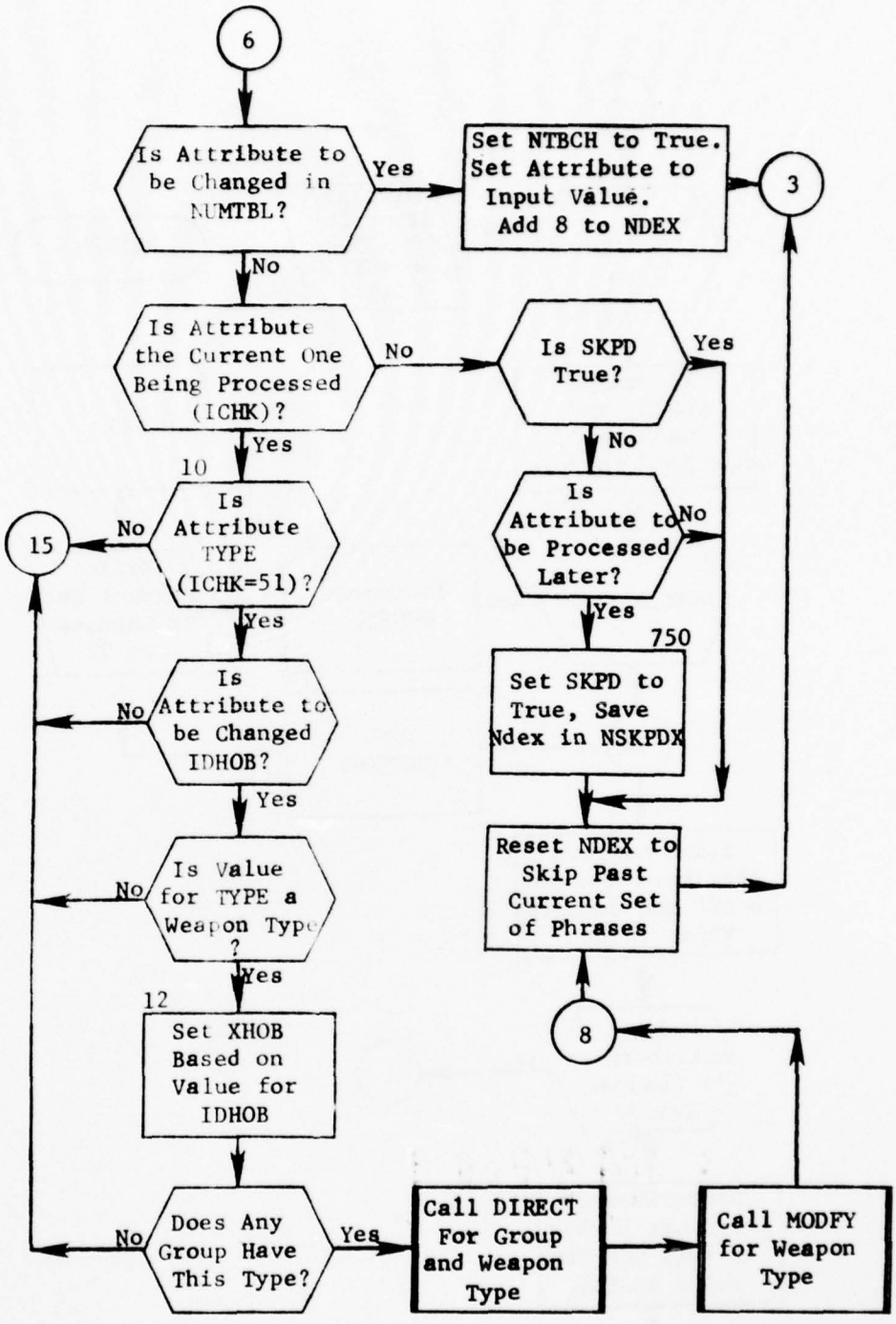


Figure 9. (Part 3 of 5)

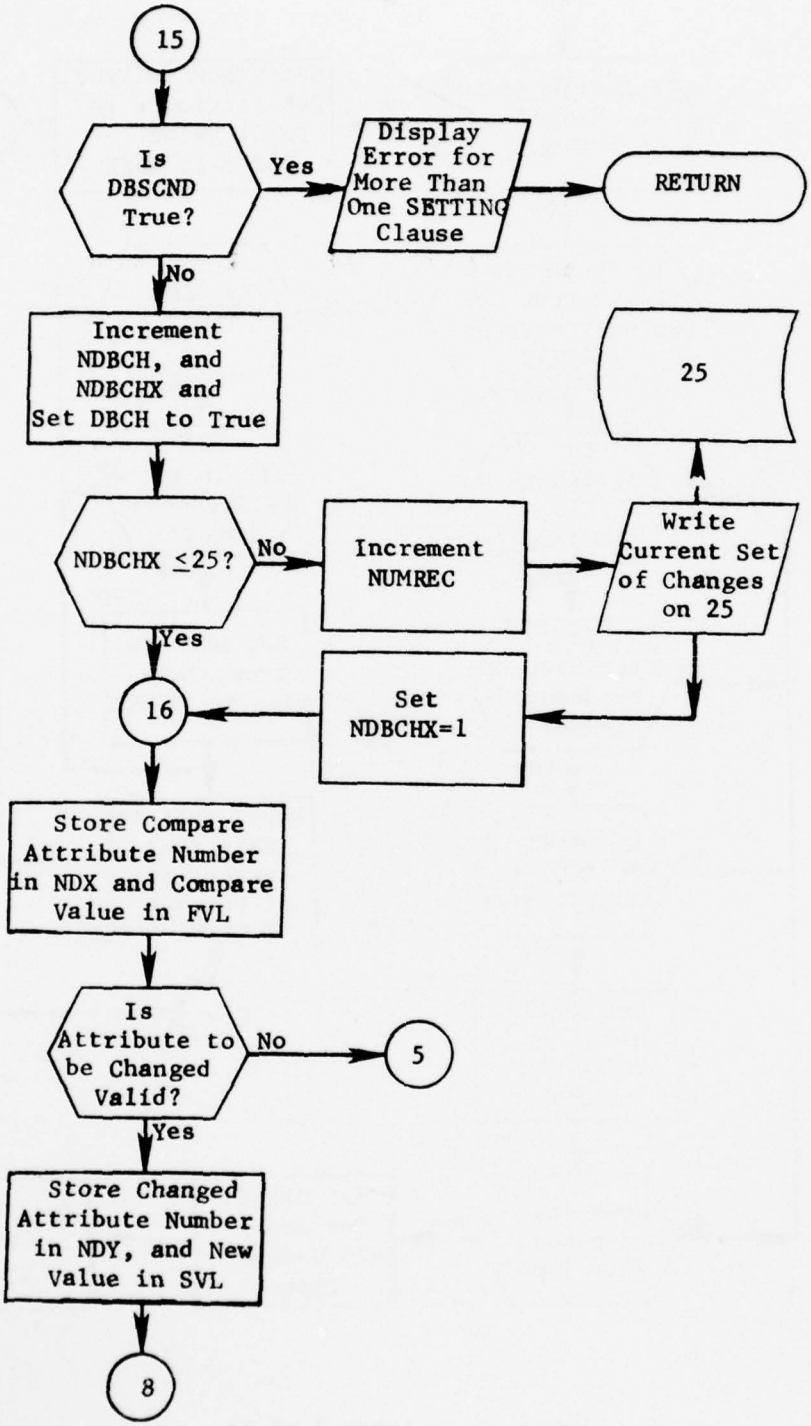


Figure 9. (Part 4 of 5)

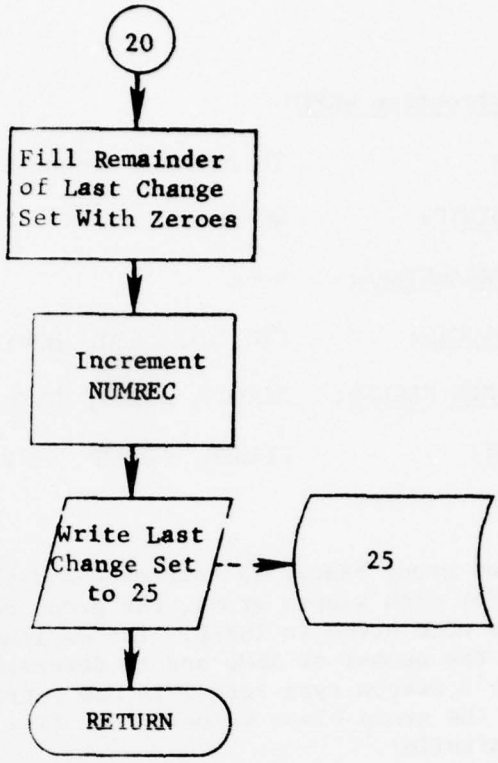


Figure 9. (Part 5 of 5)

2.14 Subroutine WEPIN

PURPOSE: To retrieve weapon data

ENTRY POINTS: WEPIN

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C15, C30, WEPCOM

SUBROUTINES CALLED: DIRECT, HDFND, HEAD, NEXTTT, RETRV

CALLED BY: FIXWEP, PRNPRP, SETRD, WEPPREP

Method:

The weapon group header is retrieved and the groups are processed in order. For each weapon group, the group record is retrieved and its reference code saved in IWRCD. The payload of the group is now scanned to count the number of ASMs and to determine if the group is a MIRV. The group's weapon type record is now retrieved and the type index set. Finally, the group class value is set (1 = missile, 2 = bomber, 3 = salvoed missile).

Subroutine WEPIN is illustrated in figure 10.

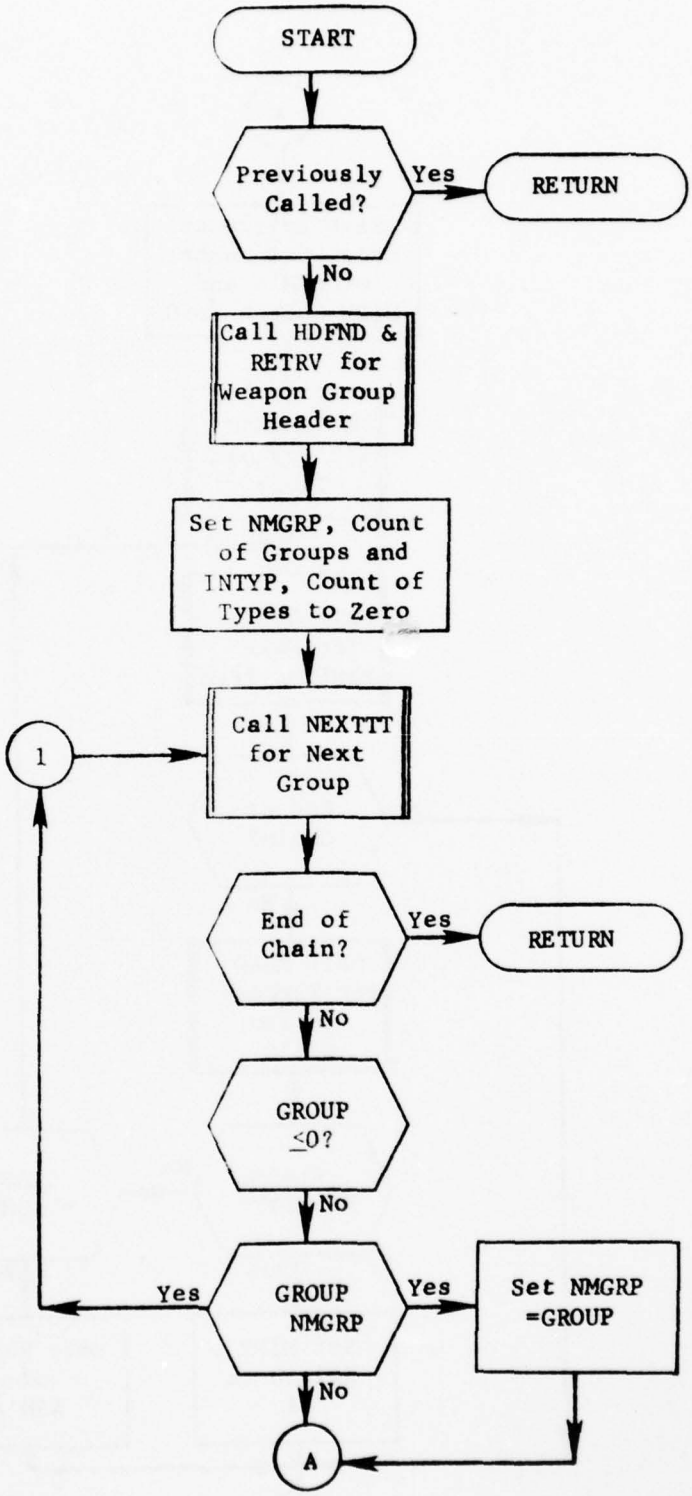


Figure 10. Subroutine WEPIN (Part 1 of 3)

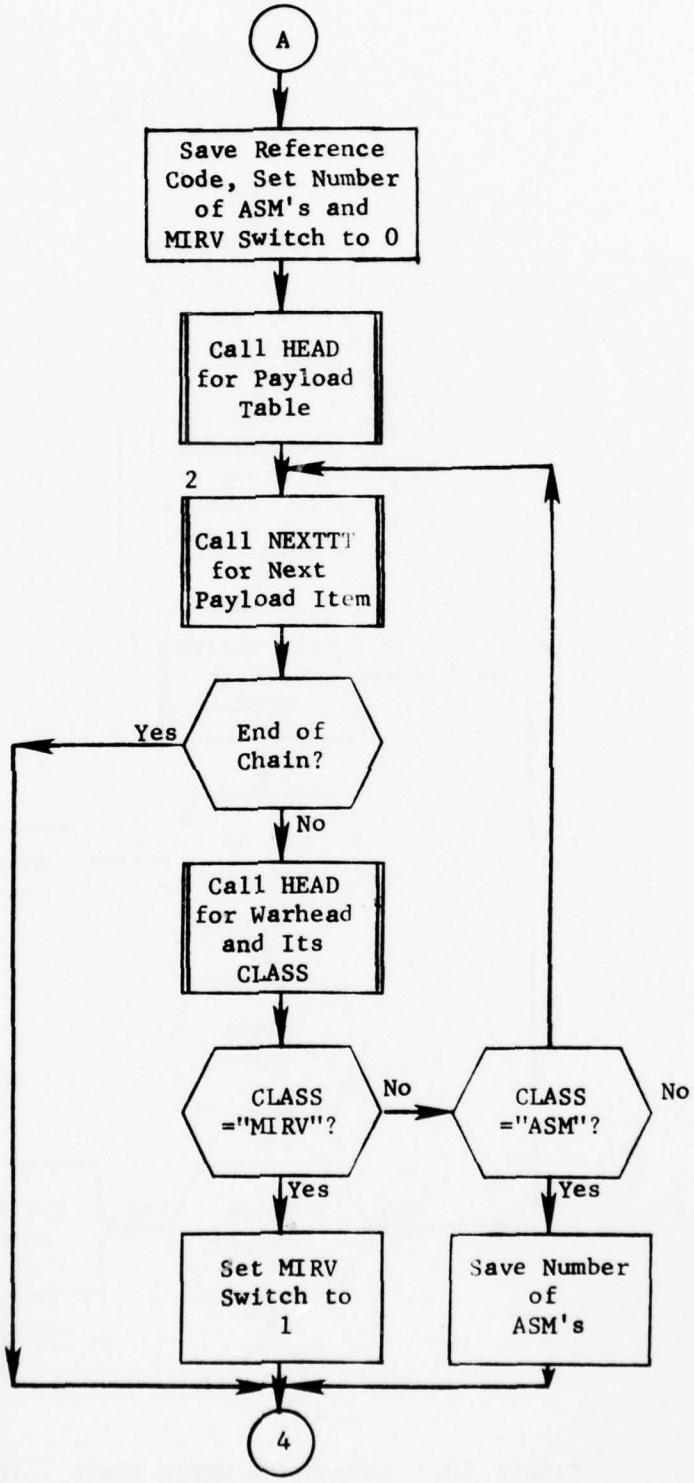


Figure 10. (Part 2 of 3)

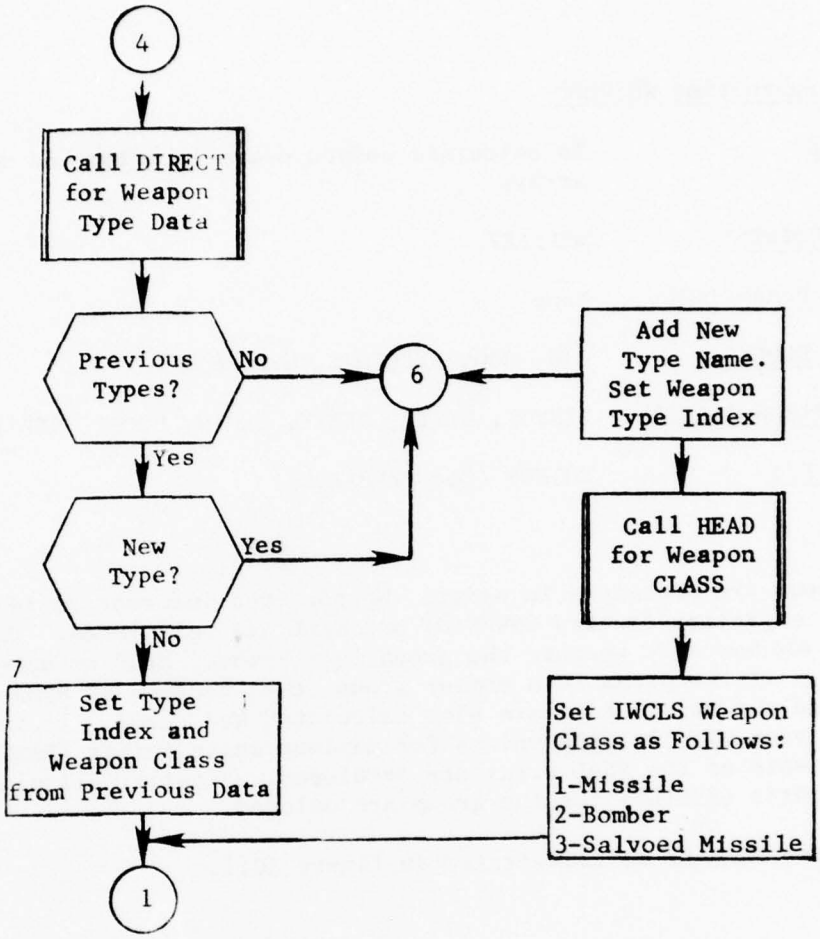


Figure 10. (Part 3 of 3)

2.15 Subroutine WEPPREP

PURPOSE: To calculate weapon overallocation and setup salvo arrays

ENTRY POINTS: WEPPREP

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C30, REPOINTS, WEPCOM

SUBROUTINES CALLED: DIRECT, DISTF, DLETE, GEOIN, MODFY, NEXTTT, WEPIN

CALLED BY: ENTMOD (for PREPALOC)

Method:

All groups are processed in order. Weapon type information is retrieved and the expansion factors GNWPNADJ and GSBL are calculated. The calculations are based on whether the group is a bomber, MIRV or non-MIRV missile. If the group is a bomber group, the fraction of ASMs in the group and the refuel time are also calculated and saved. If the group is a salvoed missile, the values for maximum salvo number (MAXSAL) and the contents of the NSAL array are developed. Finally, all old assignment records (ASSIGN) for the group are deleted.

Subroutine WEPPREP is illustrated in figure 10.1.

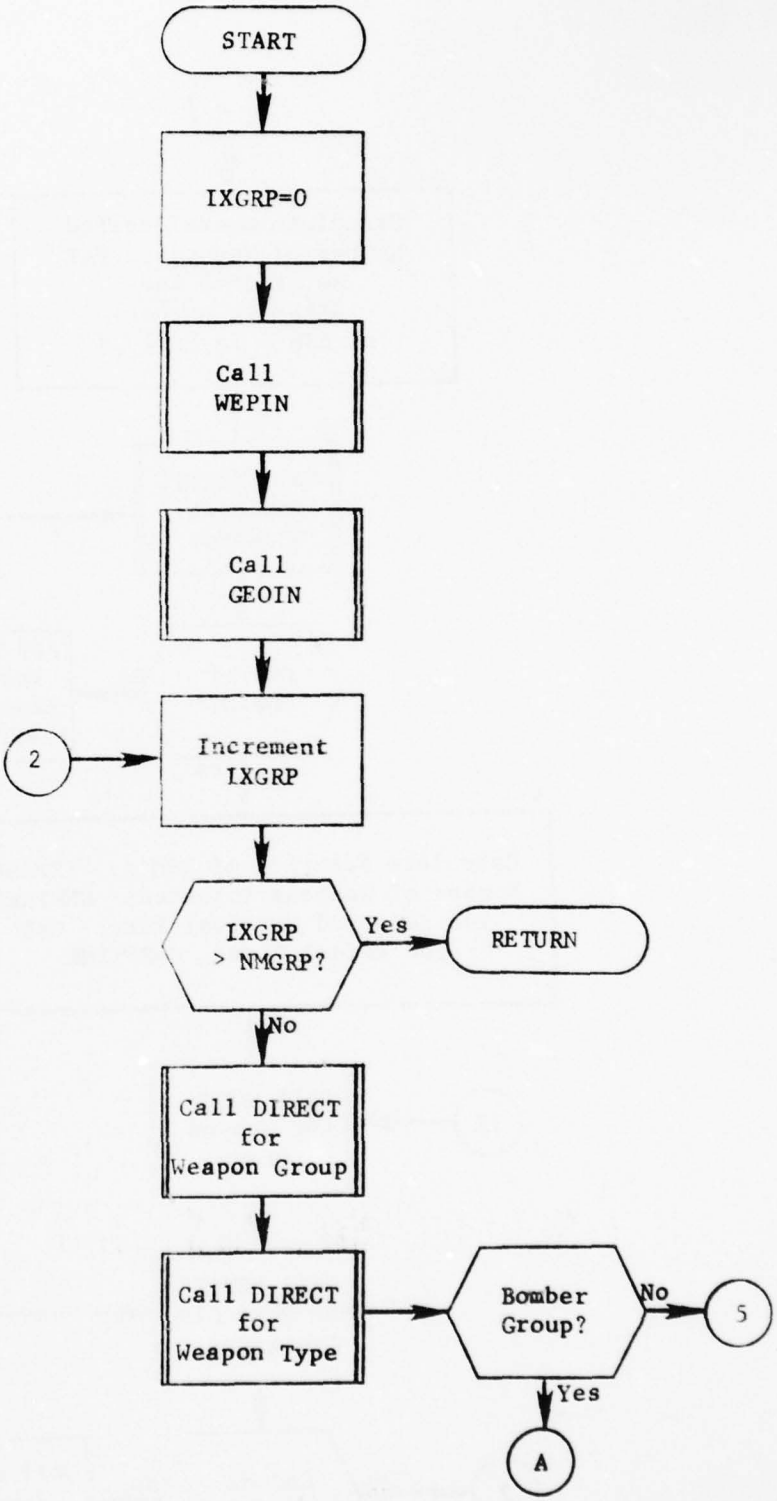


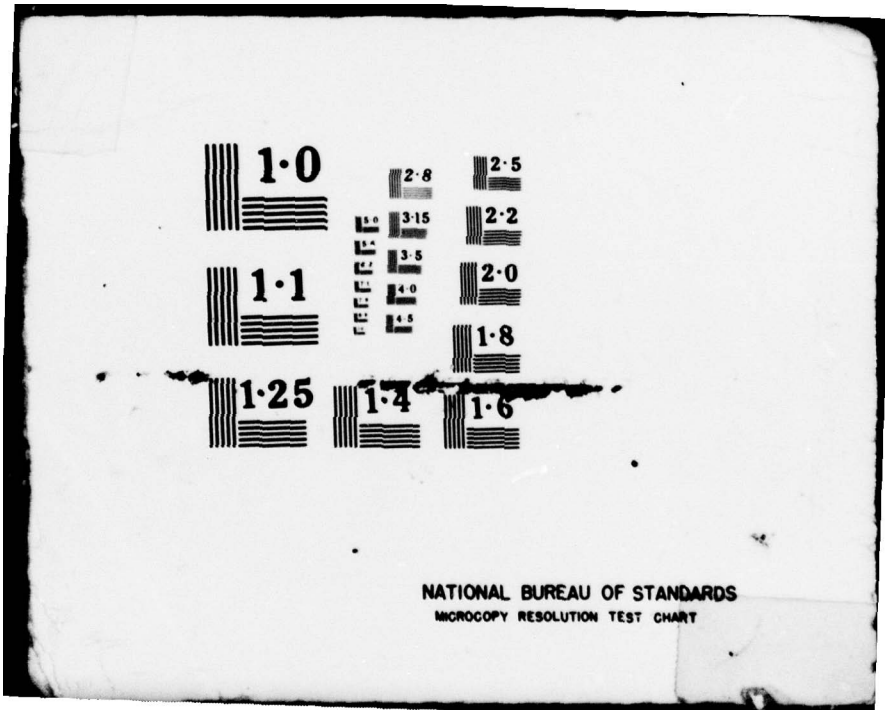
Figure 10.1. Subroutine WEPPREP (Part 1 of 3)

AD-A071 790 COMMAND AND CONTROL TECHNICAL CENTER WASHINGTON DC F/G 15/7
THE CCTC QUICK-REACTING GENERAL WAR GAMING SYSTEM (QUICK) PROGR--ETC(U)
JUN 79

UNCLASSIFIED CCTC-CSM-MM-99-77-VOL-3-C NL

2 OF 2
AD
A071790

END
DATE
FILMED
8-79
DDC


1·0
1·1
1·25
1·4
1·6
2·8
3·15
3·5
4·0
4·5
2·5
2·2
2·0
1·8
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

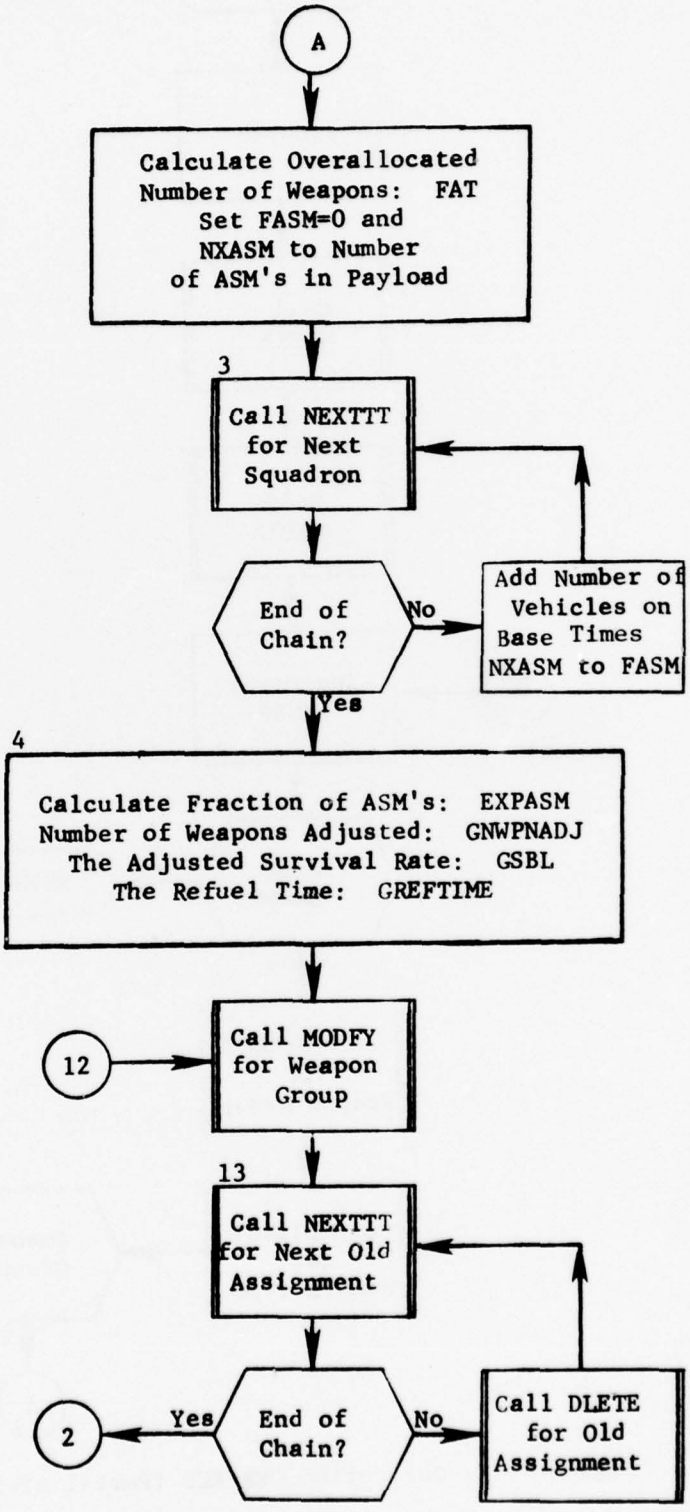


Figure 10.1. (Part 2 of 3)

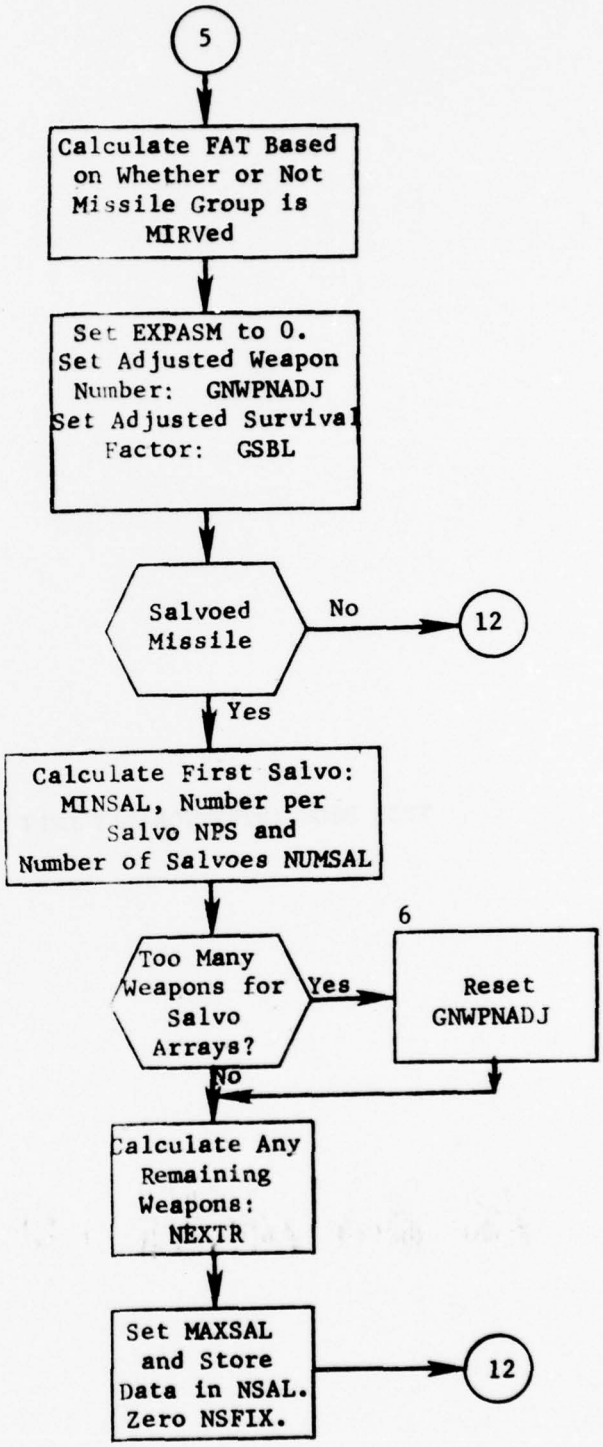


Figure 10.1. (Part 3 of 3)

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 3. ALOC MODULE

3.1 Purpose

The major purpose of this module is to determine the optimal allocation of weapons to targets, using a Lagrange multiplier technique. The weapons are divided into weapon groups -- each group containing weapons of the same characteristics which are geographically close. Thus, except for time launch interval constraints for some "salvoed" missiles, weapons are considered identical within groups. Each target is considered individually for weapon assignment. The order of investigation is the order of the TARCDE records on the LISTXX chain which was determined by the PLANSET module. When all targets have been processed, another pass over this chain begins. This process continues until the Lagrange method has allocated all the weapons to targets. The assignments are stored as ASSIGN records in the integrated data base during the process.

The user is able to specify weapon assignments through the FIX adverb to the PREPALOC module. In this case the ALOC module will optimally assign those weapons which have not been fixed. In addition, there are capabilities which allow the user to modify weapon range values, to restrict the use of MIRV weapons by target class, and to restrict the use of any weapon group by the value of either of the target attributes FLAG or CNTRYL.

3.2 Input

The precondition of the integrated data base required is that the PREPALOC module has to have been executed. Furthermore, there is an optional input file -- the Weapon/Target Data File. The Weapon/Target Data file contains the information relating each weapon group to each target. The Weapon/Target Data file, if not input, is created by the FRSTGD subroutine on pass one and may be retained for later executions of ALOC. One record is produced for each target whose length depends upon the number of weapon groups and the targets number of hardness components (length = number of groups x (3 + 2 x number of hardness components) + 1). This is the file which is created in pass one and may also be used in subsequent runs (see RECALC Mode: Users Manual, UM 9-77, Volume III). The format for the Weapon/Target Data file -- file code 15 -- appears in table 2.

3.3 Output

As a result of its execution the ALOC module creates ASSIGN (record type 70) records in the integrated data base. Further, the attribute NUMALOC is updated in every group to reflect the actual number of weapon allocated from that group.

3.4 Concept of Operation

In order to conserve storage, ALOC is broken up into two main overlays. The first overlay is called ALCINT. This overlay reads any user input,

Table 2. Format of Weapon/Target Data File -- File Code 15

STARTING WORD	DESCRIPTION
1	Target Number
2	Time of arrival of group on target
(NWEPGRP* + 2)	Corridor used by group or reason group is inactive
(2 X NWEPGRP + 2)	Penetration probability of weapon to target
(3 x NWEPGRP + 2)	Kill probability of weapon against first hardness component
(4 x NWEPGRP + 2)	Kill probability of weapon against second hardness component**
(5 x NWEPGRP + 2)***	Alternate kill probability of weapon against first hardness component
(6 x NWEPGRP + 2)	Alternate kill probability of weapon against second hardness component**

* Number of weapon groups

** Does not appear if target has only one hardness component

*** (4 x NWEPGRP + 2) -(5 x NWEPGRP + 1) if target has only one hardness component

including restrictions, modifications, print request and so on. Furthermore, the weapon data is extracted from the integrated data base. The second overlay (ALCMUL) controls the determination of the allocation. The driver routine of this overlay is MULCON. Within the second overlay there are four segments. The first, FGD, obtains target data for pass one. The second, SGD, obtains target data for passes two and beyond. The third segment, STAL, principally routines STALL, WAD and WADOUT, allocates to targets without terminal ballistic missile defenses. The fourth segment, DEFAL, principally routines DEFALOC and RESVAL handles ballistic missile defended targets.

3.4.1 Overlay ALCINT. The routines in this overlay are straightforward and need little explanation beyond that below. However, the routine DATGRP places information on a sequential file (file code 23) which is used by the FGD segment. Table 3 shows the format and content of this file.

3.4.2 Overlay ALCMUL. The design of the weapon-to-target allocator utilizes a hierarchy of subroutines operating at different levels of detail. Figure 11 illustrates this hierarchy. The major functions associated with these subroutines are summarized below and related to the overall concept in subsequent paragraphs.

Subroutine MULCON is the first subroutine in the hierarchy and is responsible for the control and adjustment of the Lagrange multipliers. MULCON monitors the rate at which various classes and types of weapons are being allocated to the target system and makes appropriate adjustments in the values of the Lagrange multipliers. In this role, MULCON does not need any detailed information concerning actual allocation. It is concerned only with the actual rate of allocation of the available inventory as the targets are processed. To obtain the assignment of weapons to each successive target, MULCON simply calls subroutine STALL (Single Target Allocator) for targets without missile defenses, or subroutines STALL and DEFALOC if the target is defended. STALL and DEFALOC utilize the current values of the multipliers to make an allocation to the next target, then return control to MULCON.

The data acquisition for the allocation process is performed by the FRSTGD routine on pass one and SCNDGD on all other passes. Each of these routines brings in the proper IDS records for the next target and prepares the weapon data for that particular target. The Weapon/Target Data File is read (or on the first pass in the RECALC mode calculated). On the first pass, FRSTGD then creates a record on file code 21 which, principally, contains the INACTIVE array (see table 4). If the user has requested range modification, FRSTGD may also write a record onto file code 22 in the same format as file code 15 (table 2). This new file serves as a source for replacement record for file code 15. SCNDGD reads each of these files in order to obtain the appropriate information.

Table 3. Weapon Data File from DATGRP

WORD	DESCRIPTION
1	Group type index number
2-4	Logical flag restrictions (99 flags packed 1 bit per flag)
5-9	Logical country location restriction (150 switches corresponding to codes in block CNCLS, packed 1 bit per switch)
10	Switch -- true if group is restricted MIRV
11	Logical MIRV restriction switches (packed 1 bit per switch)
12	Range multiplier
13	Refueled range multiplier
14	Minimum range replacement value
15	NALTDLY for group
16	ALTDLY for group
17	GLAT for group
18	GLONG for group
19	GREFCODE for bomber group, 0 for missile group
20	GYIELD for group
21	RANGE for group
22	CEP for bomber group, CEP at 0 range for missile group (CEP = RANGE * (CEP-CEPMIN)/(RANGE-RANGMIN))
23	SPEED for group
24	RANGED for group
25	RANGER for group (in DATAMAKE mode, PENPROB)
26	RNGMIN for group
27	GREFTIME for group
28	TOFMIN for group
29	CMISS for group
30	Slope of CEP equation for missiles, 0 for bombers

Table 5. (Part 2 of 10)

BLOCK	ARRAY OR VARIABLE	DESCRIPTION
CORSTF (cont.)	ORGLAT(30)	Origin latitude
	ORGLONG(30)	Origin longitude
	DISTCDZ(30)	Distance from origin to target
	ATTRAD(30)	Sum of precorridor and corridor attrition
	CROSSDST(30)	Perpendicular distance from axis to target
	ENTLAT(30)	Entry latitude
	ENTLONG(30)	Entry longitude
	DISTDG	Distance from target to recovery base
	MAXCOR	Number of corridors
CURSUM	CSALL	Number of targets assigned to all weapons
	CSREG(5)	Number of targets assigned to weapons from a given region
	CSCLAS(2)	Number of targets assigned to weapons from a given class
	CSTYPE(120)	Number of targets assigned to weapons of a given type
	CSGRP(250)	Number of targets assigned to weapons from a given type
	CSOTH(2)	Number of targets assigned to weapons with a given alert status
DEFCOM	RATM	Highest return rate from DEFALOC
	ISALFX(250)	Storage for fixed salvo numbers
	NSL(250)	Number of salvoed weapons available
	RATE(250)	Rate of return for missile on defended target
DEFRES	NOWEP(250)	Number of weapons assigned by DEFALOC
	VTDX	Surviving target value
	NTX(3)	Terminal defender estimates
	PX(3)	Probability of NTX

Table 5. (Part 3 of 10)

BLOCK	ARRAY OR VARIABLE	DESCRIPTION
DYNAMIC		Contains allocation
	IG(30)	Group assigned
	KORRX(30)	Corridor assigned
	RVALX(30)	Relative value of assignment
	PENX(30)	Penetration probability
	TOARR(30)	Time of arrival
	ISAL(30)	Salvo number
	NUMFIX	Number of fixed assignments
	NUM	Number of assignments
FIL21	JTG	Target number
	ILENTH	Length of record on file 15 (or 22)
	I22SW	Indicates need to read file 22
	ICTIVE(250)	Storage for INACTIVE array
FIRST	FIRST	Indicates first target of pass
	END	Indicates end of target list
	F22LSW	Indicates file 22 in use
FLGSTF	LFLAG(688)	Packed logical flag restrictions
	LCNTRY(1042)	Packed logical location restrictions
	NMMRV	Number of restricted MIRV groups
	MRNAM(100)	Payload table name of restricted MIRV
	LCLAS(67)	Packed logical MIRV class restrictions
	RNMUL(250)	Range multiplier
	RNRMUL(250)	Refueled range multiplier
	RNMIN(250)	Range minimum replacement
FORMTT	INWORD	Value which needs a format
	NFORMAT	Format for INWORD
GRPHDR	IWGHDR	IDS Reference Code for weapon group header

Table 5. (Part 4 of 10)

BLOCK	ARRAY OR VARIABLE	DESCRIPTION
GRPSTF		Contains record from file 25
	ITYPE	Group's type index
	LFLAG (3)	Flag restrictions
	LCNTRY (5)	Location restrictions
	LMRSW	Indicate if group is restricted MIRV
	LCLASS (1)	MIRV restrictions
	RMUL	range multiplier
	RRMUL	Refueled range multiplier
	RMIN	Minimum range replacement
	GPARAM (16)	Contains the following group parameters in order: NALTDLY, ALTDLY, GLAT, GLONG, GREFCODE, GYIELD, RANGE, CEP, SPEED, RANGED, RANGER, RNGMIN, GREFTIME, TOFMIN, CMISS, SLOPE
INITSW	RECALC	Indicates RECALC mode
	PUNSW	Indicates output of final lambda's is desired
	FLAGSW	Indicates flag restrictions
	LOCRSW	Indicates location restrictions
	RMODSW	Indicates range modifications
	MRVRSW	Indicates MIRV restrictions
	PUNIT	Logical unit on which final lambdas are to be output
LACB	LALL	Lambda for all weapons
	LAREG(5)	Lambda for a given region
	LACLAS(2)	Lambda for a given class
	LATYPE(120)	Lambda for a given weapon type
	LAGRP(250)	Lambda for a given group
	LAOTH(2)	Lambda for a given alert status
MULTIP	CTMULT	Current target multiplicity
	NSPLITS	Number of splits in current multiple target

Table 5. (Part 5 of 10)

BLOCK	ARRAY OR VARIABLE	DESCRIPTION
MULTIP (cont.)	ISPLIT	Index of current split
	NSPREC	Index of file 25 record used for this target
	KLMULT	Split range indicator
NALLY	NALL(250)	Number from group allocated on this pass
	RNALL(250)	Number from group currently allocated
NOWPS	NOALL	Number of weapons total
	NOREG(5)	Number of weapons in a given region
	NOCLAS(2)	Number of weapons in a given class
	NOTYPE(120)	Number of weapons of a given type
	NOGRP(250)	Number of weapons in a given group
	NOOTH(2)	Number of weapons with a given alert status
PAYOFF	OPROFIT	Profit from old allocation
	SPAYOFF	Sum of all payoffs
	SUMCOST	Sum of all costs
	SPROFIT	Sum of all profits
PAYSAV	GSCC(100)	CCREL for payload
	GSREL(100)	REL for payload
	GSEASM(100)	EXPASM for payload
	GSLINT(100)	LCHINT for payload
	NGSWHD(100)	NWHDS for payload
	NGSDEC(100)	DECOYS for payload
	GPAYALT(100)	PAYALT for payload
	GYLDASM(100)	ASM yield for payload
	IWHOB(100)	Payload height of burst 0 = for ground 1 = for air 2 = if not preset

Table 5. (Part 6 of 10)

BLOCK	ARRAY OR VARIABLE	DESCRIPTION
PNAV	GSPKNAV(100)	PKNAV for payload
PREMS	PREMIUM(250)	Premium for using weapon
	DPREMIUM(250)	Premium for deleting weapon
	SUMPREM	Sum of premiums
	TBENEFIT	Total benefit
PRNTCN	IDO(40)	Print request activation switch
	INDEXPR(40)	Print request selection number
	JPASS(40)	Print request first pass
	JTGTP(40)	Print request first target
	LPASS(40)	Print request last pass
	LTGT(40)	Print request last target
	KTGTFREQ(40)	Target print frequency
	ICOUNT(40)	Print request frequency counter
	MAXREQ	Maximum number of requests
	MPRNT	Number of array entries
	NREQ	Number of requests
PRTMUL	PROCMULT	Current fraction of multiple target class
	DELTEFF	Increase in profit/VALWPNS
	SDELTEFF	Sum of DELTEFF
	VALWPNS	Sum of all weapon values (lambdas)
	VALERR	Value of surplus plus deficit weapons
REFPNT	RFLAT(10)	Refuel point latitude
	RFLONG(10)	Refuel point longitude
SALVO	NSALW	Number of salvoed weapon groups
	MXSAL(75)	Maximum salvo number per weapon group
	NSALAL(450)	Running sums of salvo allocation (six words per salvoed group - packed four sums per word)
	LXIHAVE(50)	Packed logical switch indicating salvo with weapons

Table 5. (Part 7 of 10)

BLOCK	ARRAY OR VARIABLE	DESCRIPTION
SALVO (cont.)	SAVLAM(250)*	Storage for salvoed weapon lambdas (contains average payload difference for bombers - AVDE)
	MYSAL(250)*	Available salvo (contains bomb/ASM selection for bombers - ISETPAY)
	P(250)*	Balance parameter (contains current utilization of ASMs for bomber - FASM)
	ISALPT(250)	Salvo index, indexes arrays MXSAL, NSALAL and LXIHAVE: 0 for nonsalvoed groups
SPLITS	ZZ(203)	Buffer for file 25
	SPLTMD	Switch to indicate data modified since last read
	NBLNX	Number of splits
	TMX	Not used
	INDEX	File 25 index
	STARG(3)	Starting target numbers of split
	IOFF	Offset of data in ZZ
	NTOTGT	Number of targets - all splits
	SPDAT	Not used
SMATAD	SMNOMIRV(3)	SMAT parameters for non-MIRVs
	SMATMIRV(3)	SMAT parameters for MIRVs
SURPW	SURPWP(250)	Estimated weapon surplus
TABLE	TABLE(101)	Table of square root law K-factors
TGTSAV	TGTLAT	Target latitude
	TGTLONG	Target longitude
	TGTCLS	Target class name
	VO(2)	Target value per hardness component
WADFIN	VTP(250)	Value remaining at target after weapon added
	DELVT(30)	Difference in surviving value
	NUMO	Numbers of old allocations
	IGO(30)	Group numbers of old allocation

* For bomber groups these arrays are equivalenced to arrays AVDE, ISETPAY and FASM.

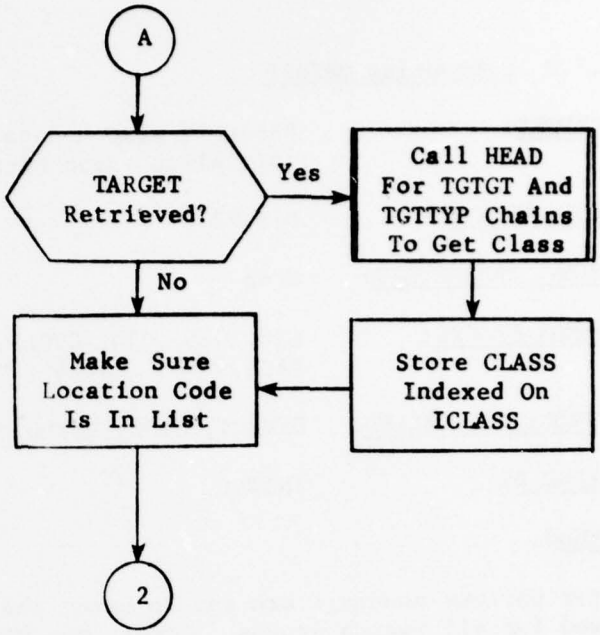


Figure 17. (Part 2 of 2)

3.7.2 Subroutine DATGRP

PURPOSE: Assemble weapon data, build payload tables and initialize salvo arrays

ENTRY POINTS: DATGRP

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C15, C30, CNCLS, FLGSTF, GRPHDR, INITSW, LACB, NOWPS, PAYSAV, PNAV, SALVO, WEPSAV, WPFIX

SUBROUTINES CALLED: DIRECT, GLOG, HDFND, HEAD, NEXTTT, RETRV, SLOG

CALLED BY: INITAL

Method:

After various counters are set to zero, the following process is followed for all weapon groups. First, the TYPE and CLASS are determined and some group attributes are saved. Next the payload for the group is examined and compared with the payloads of previous groups. New payloads are stored and old payloads are indicated via the index in array IPAY. Next the values for file 23 are determined. This file contains any flag, location or MIRV restrictions, range modifications and group parameters that will be required on the first pass. Next, the weapon totals in block NOWPS are updated and the group lambda calculated. Finally, if this is a salvoed group, the salvo arrays are initialized. This final process includes the unpacking of NSAL (9 weapon salvos per word) and repacking into NSALAL (4 weapon salvoes per word).

Subroutine DATGRP is illustrated in figure 18.

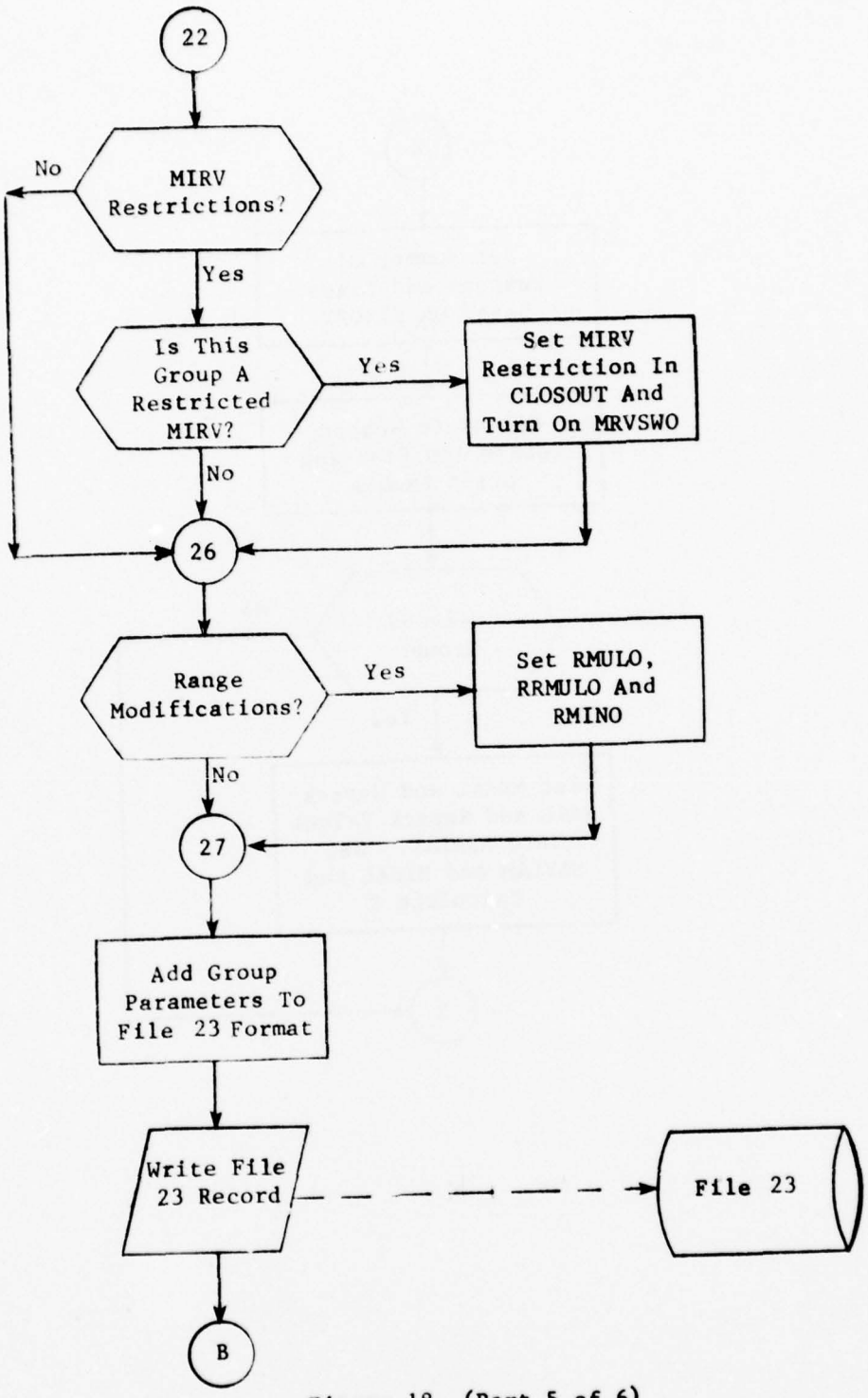


Figure 18. (Part 5 of 6)

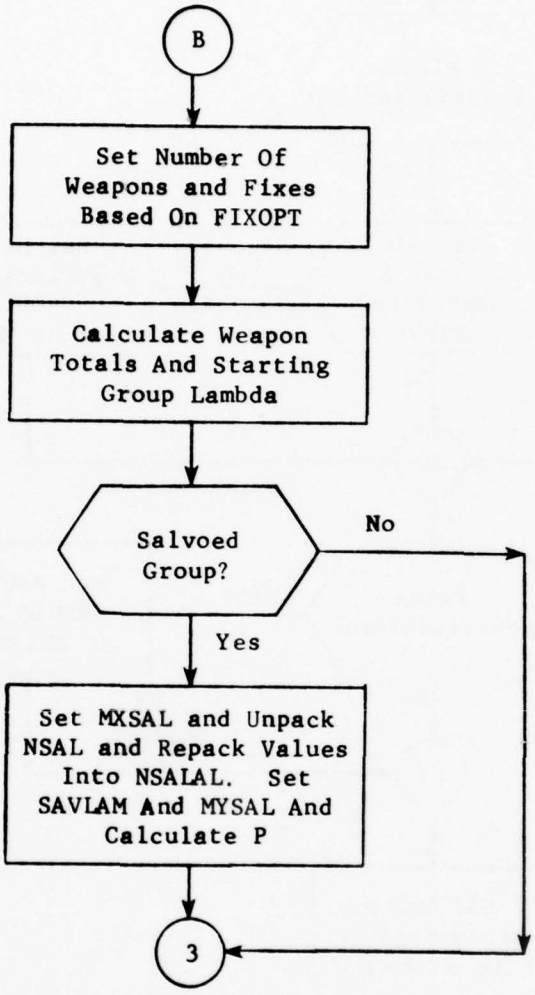


Figure 18. (Part 6 of 6)

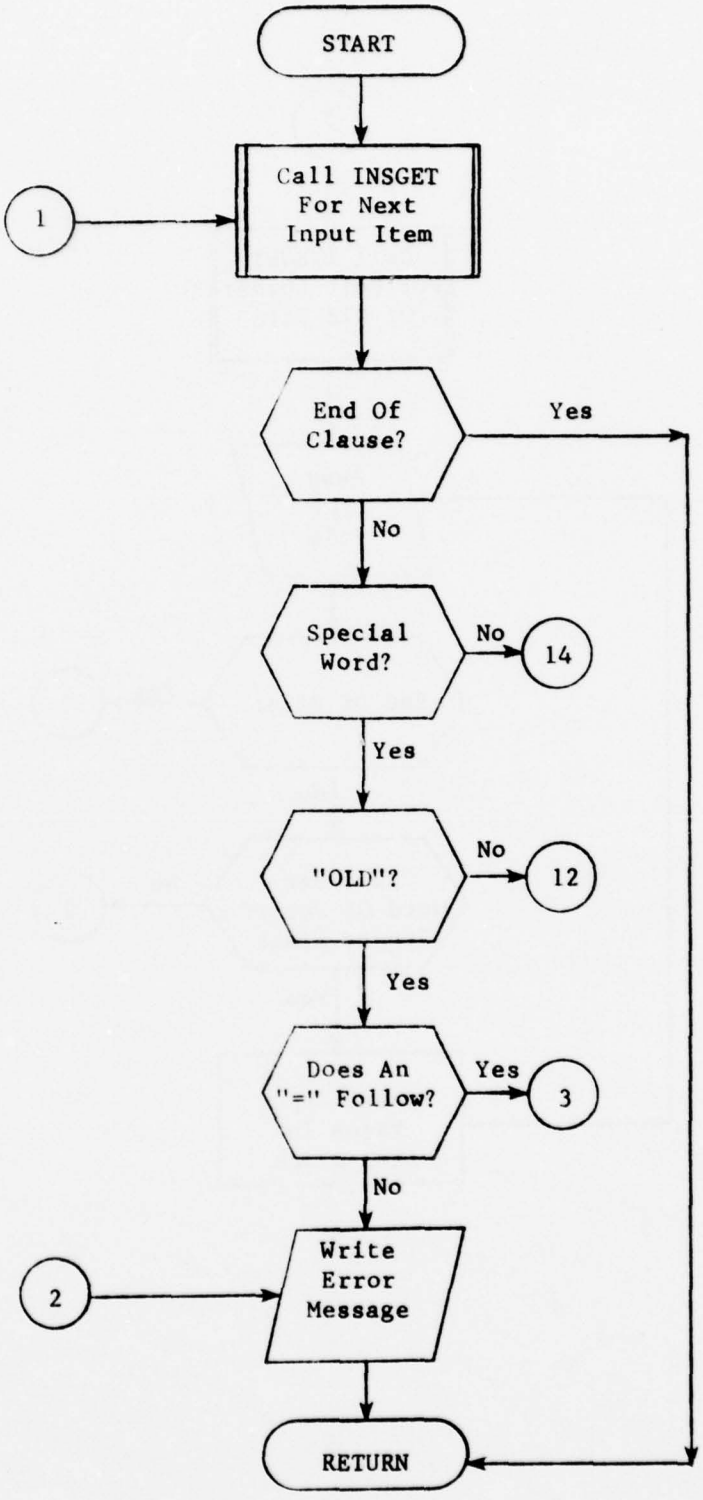


Figure 22. Subroutine RDMUL (Part 1 of 5)

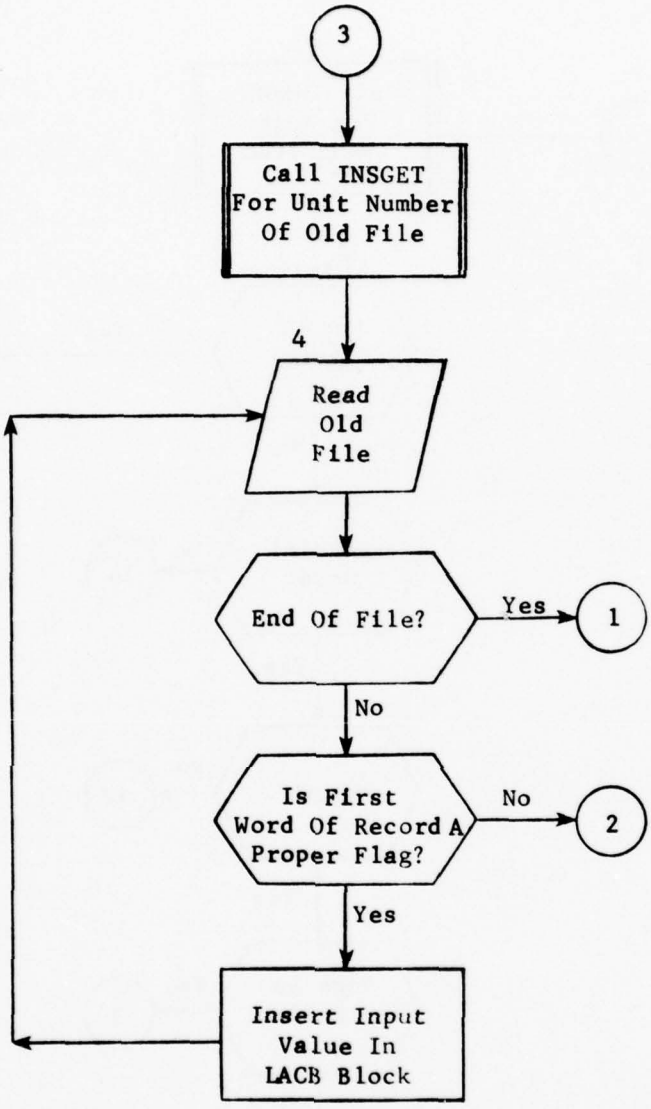


Figure 22. (Part 2 of 5)

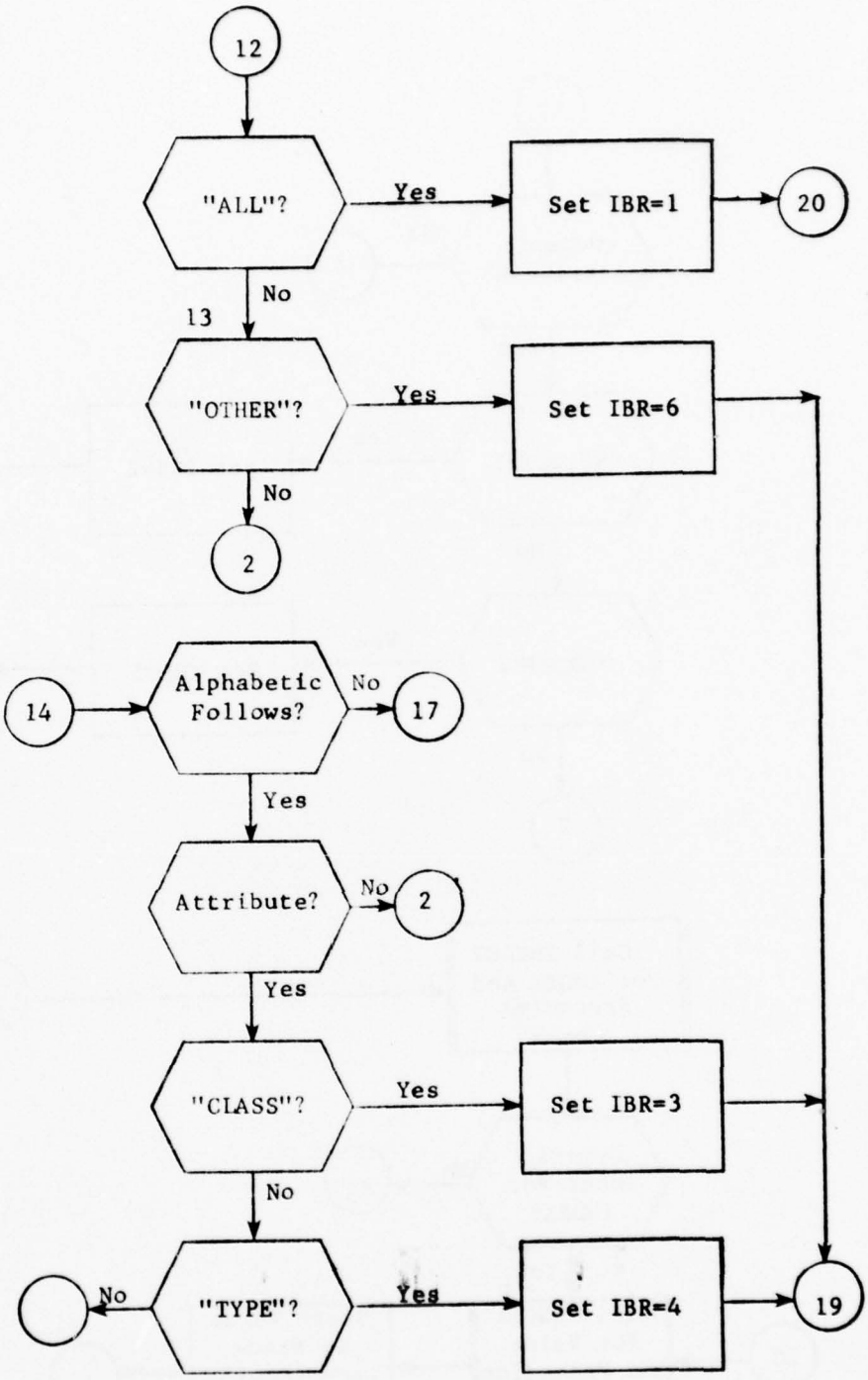


Figure 22. (Part 3 of 5)

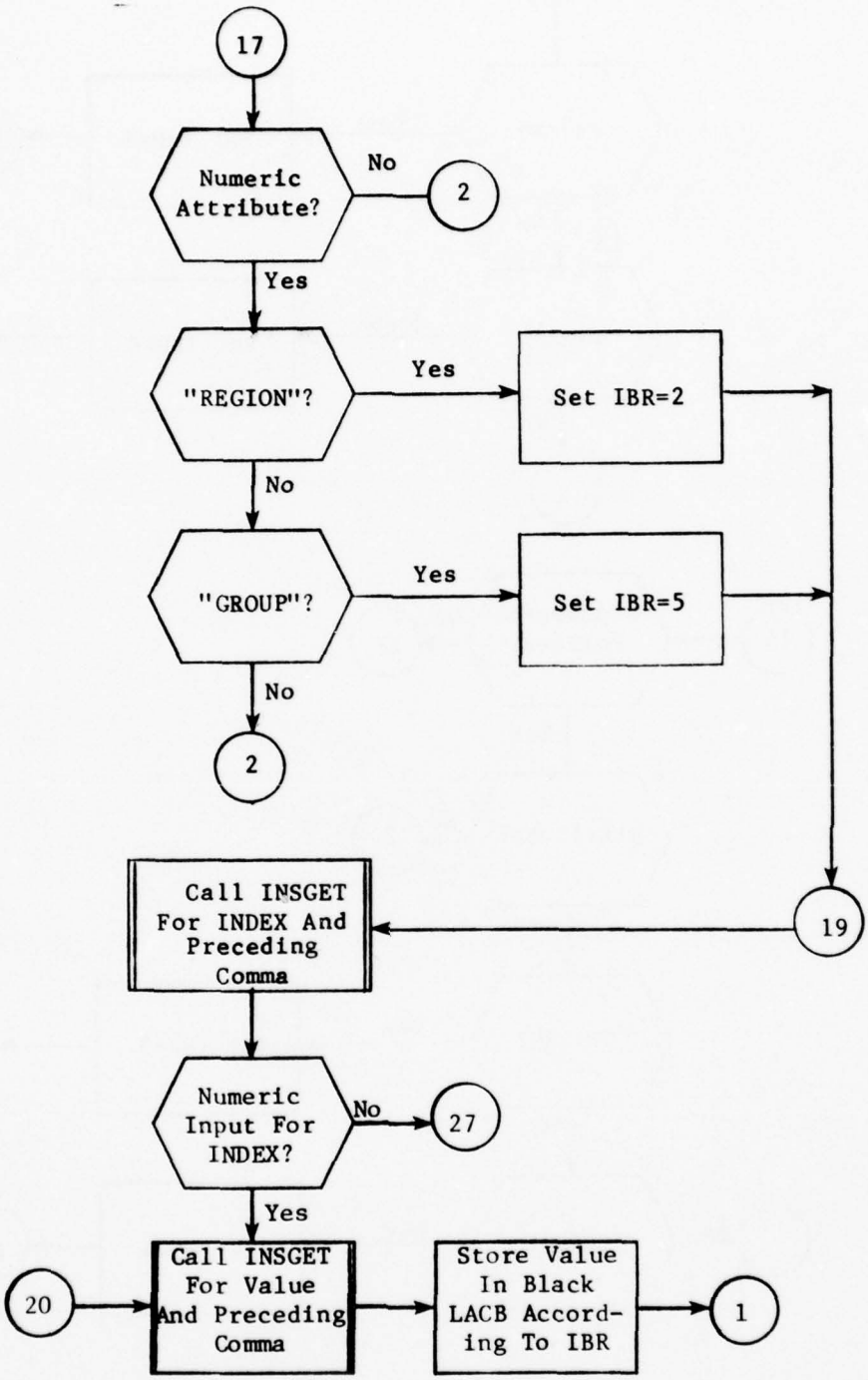


Figure 22. (Part 4 of 5)

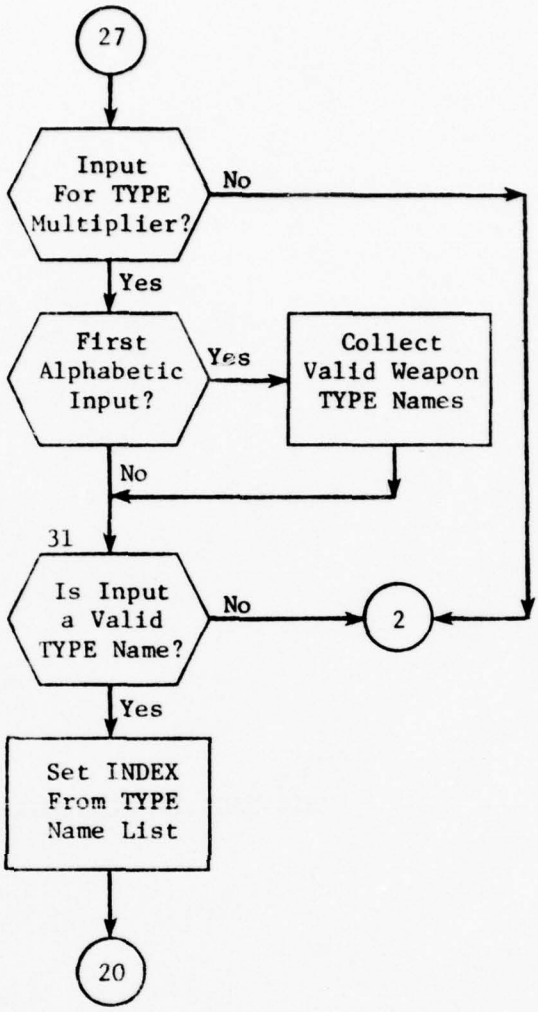


Figure 22. (Part 5 of 5)

THIS PAGE INTENTIONALLY LEFT BLANK

allocation; consequently a revised cost (REVCOST) of the allocation based on the new multipliers is of interest, and is probably different than the old cost COST. The reason for the test on PROGRESS before correcting the cumulative differential profit will be discussed in connection with Part IV of the program (part 5 of figure 27).

Subroutine ADDSAL is used to maintain the sums for the salvoed groups.

Part IV: Processing After Allocation

Before calling STALL,* CTSPILL is set to 0. (If some elements are spilled, WAD will so note by setting CTSPILL equal to the number of elements spilled.) The calls on TIMEME (5, 6, 7) before and after the allocation cause the time spent during the actual allocation to be recorded in columns 6 and 7 of the TIMEME output print (number 23).

Before calling either allocation routine, however, the program must check the number of fixed weapon assignments. The limitation of weapons allocated to one target is 30 weapons on an undefended target and 30 weapon groups on a defended (i.e., terminal ballistic missile defenses) target. Usually, MULCON calls both STALL* and DEFALOC* on defended targets and chooses the best allocation. If there are more than 30 fixed weapon assignments, STALL* should not be called. If the number of fixed assignments is greater than 30, MULCON checks to see if it is a defended target. If not, an error message is printed and the excess assignments are ignored. Then STALL* is called. If it is a defended target, MULCON sets a dummy low profit (except for verification) and calls only DEFALOC.*

The additional details of the allocation required by later processors are then recorded in /DYNAMI/. PENX and TOARR are required by EVALALOC, while KORRX and RVALX are required by ALOCOUT, FOOTPRNT, and POSTALOC. Subroutine BOMPRM is then called to update the ASM allocation fraction array FASM.

The various running sums are then calculated. If DEFALOC has made the allocation, the KORRX array gives the number of missiles from each group allocated to the target. If KORRX is positive, it represents the corridor. If it is negative, it represents the number allocated. Then the profit and cost data are recorded.

The following quantities are of particular importance:

DPROFIT = PROFIT - OPROFIT

SDPROFIT = Σ DPROFIT

DELTEFF = DPROFIT/VALWPNS

SEDLTEFF = SDPROFIT/VALWPNS

* STALL and DEFALOC are called via computer system subroutine LLINK.

These quantities are computed and the last two printed out in the standard ALOC print number 2 to help the user evaluate the progress of the allocation.* The quantity OPROFIT represents the profit of the old allocation to the target evaluated in terms of the present values of the Lagrange multipliers. DPROFIT is thus a measure of the improvement in profit using the new allocation. Until PROGRESS = 1.0 this quantity is summed over all targets (one complete pass only) to give SDPROFIT. Thus, when the multipliers have been near the correct values for one full pass the value of SDPROFIT should be small. To provide a standard relative value for interpreting these quantities, they are divided by the value of all the weapons VALWPNS,

$$\text{VALWPNS} = \sum \text{NWPNS}(G) * \text{LAMEF}(G)$$

to obtain DELTEFF and SDELTEFF which measure changes in profit as a fraction of the total value of all weapons.

The quantity of SDELTEFF, therefore, provides an estimate of how efficient the allocation would have been if the allocation had been terminated one pass earlier. Presumably, the current efficiency is substantially higher, but SDELTEFF does not, at this point, give any indication of how much. It is nevertheless of value in developing experience on how soon the PROGRESS .75 phase can be terminated. When PROGRESS is equal to 1.00 the multipliers are frozen and this role of SDELTEFF ceases to be relevant. The quantity is then reset to 0. Thereafter it provides a measure of the effect on the profit of closing to the exact stockpile. Usually during the closing phase SDELTEFF goes slightly negative. However, since during this phase we continue to replace allocations originally produced with slightly different values of the multipliers, the value may go positive for a while until the closing forces get large enough to force closure even at some loss of profit. Thus the value of SDELTEFF at the end of the closing phase (PROGRESS = 1) measures the loss in profit associated with closing. In the event that closing requires more than one full pass, a test has been inserted which causes SDELTEFF to continue to accumulate over more than one pass when PROGRESS = 1.0.

Finally when PROGRESS = 2.0 the quantity is again set equal to 0. If a verification pass is carried out, SDELTEFF then measures any increase in profit in the verification pass relative to the final allocation. In this role it defines an upper limit on the inefficiency of the actual allocation.

Ordinarily after all these calculations are performed ASGOUT is called to store the results. However, if PROGRESS = 2 this step is skipped since the integrated data file already contains the final allocation.

---

* The column labelled (P-O)/VWPS in print number 2 contains these variables: DELTEFF on the first line, SDELTEFF on the second.

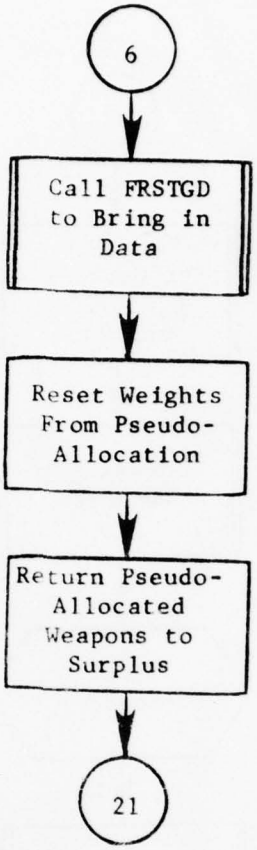


Figure 29. Part II: First Pass Processing

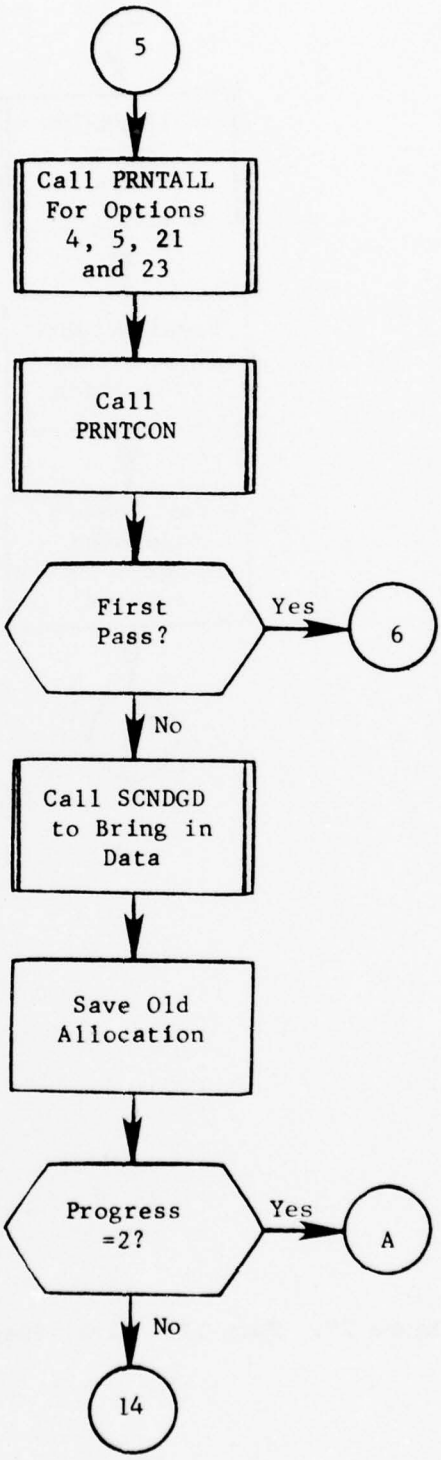


Figure 29. Part III: Main Flow (After First Pass) (Part 1 of 3)

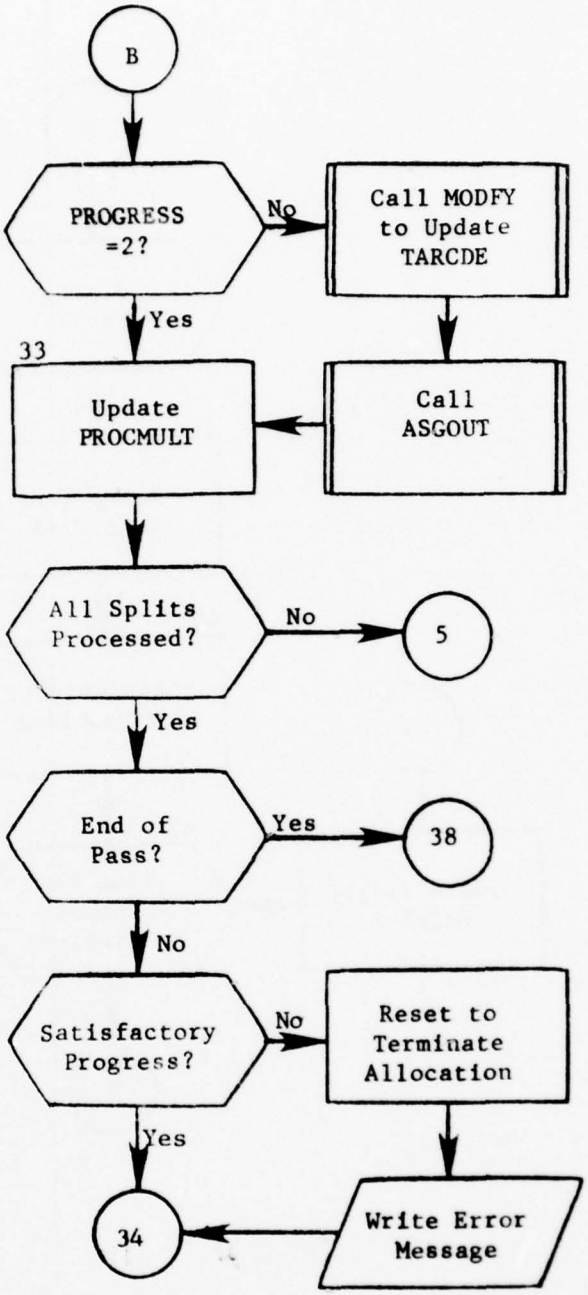


Figure 29. Part IV: (Part 3 of 4)

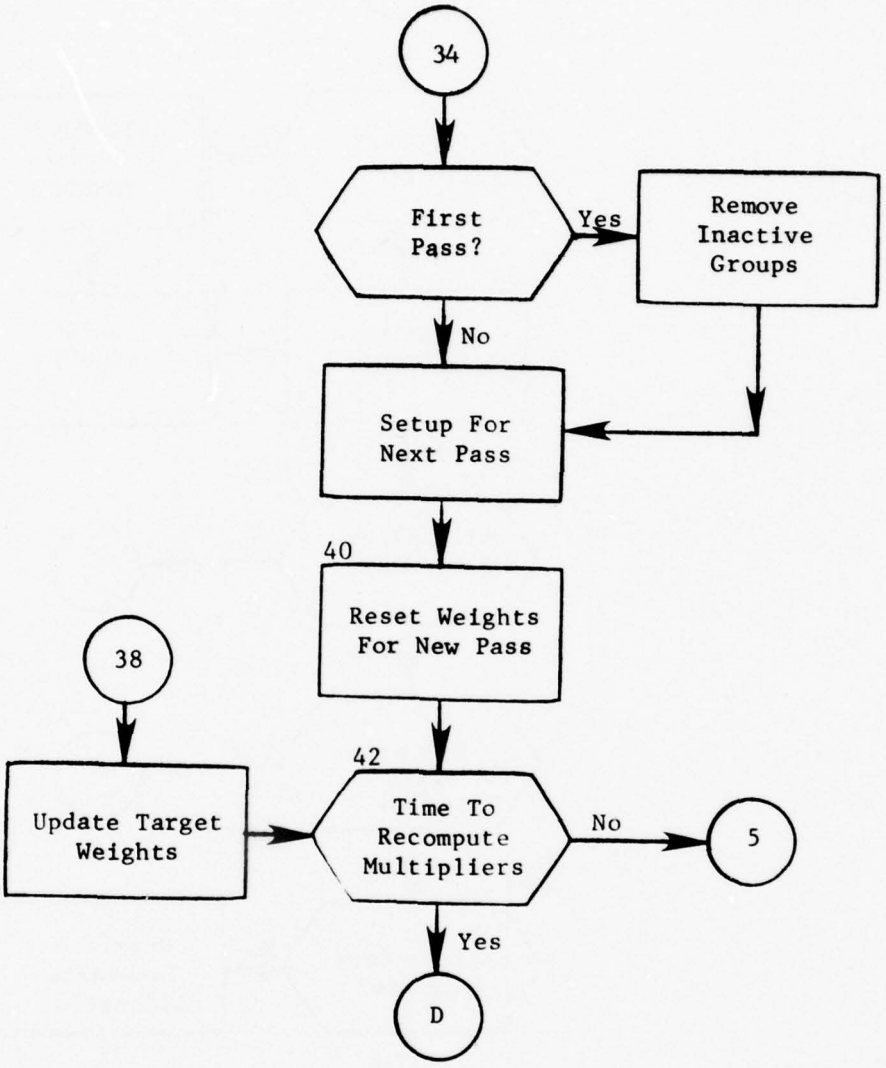


Figure 29. Part IV: (Part 4 of 4)

3.8.2 Subroutine ASGOUT

PURPOSE: To update allocation assignment records in the integrated data base

ENTRY POINTS: ASGOUT

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C30, DYNAMI, MULTIP, SPLITS, PAYSAV, TARREF, TGTSAV, WEPSAV

SUBROUTINES CALLED: DIRECT, DLETE, MODFY, MYAPOS, NEXTTT, STORE

CALLED BY: MULCON

Method:

First a logical switch is set for each new weapon assignment to indicate it is unassigned. Next each old assignment is compared to the new assignments to see if all values, save RVAL, of the old assignment are equal to a new assignment. If it does, the RVAL attribute is deleted. Finally, an ASSIGN record is created for all the new assignments for which there is no match.

Subroutine ASGOUT is illustrated in figure 31.

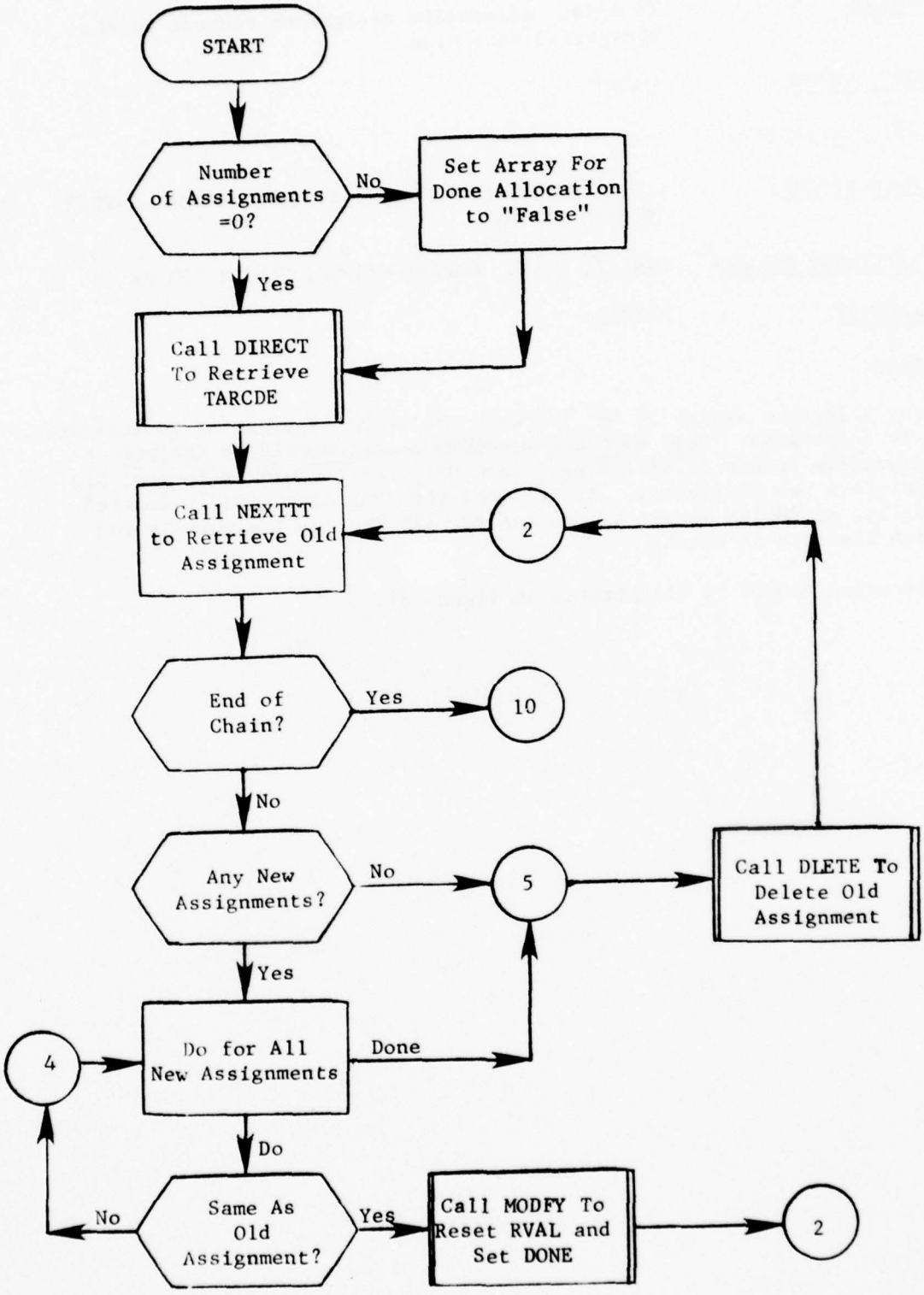


Figure 31. Subroutine ASGOUT (Part 1 of 2)

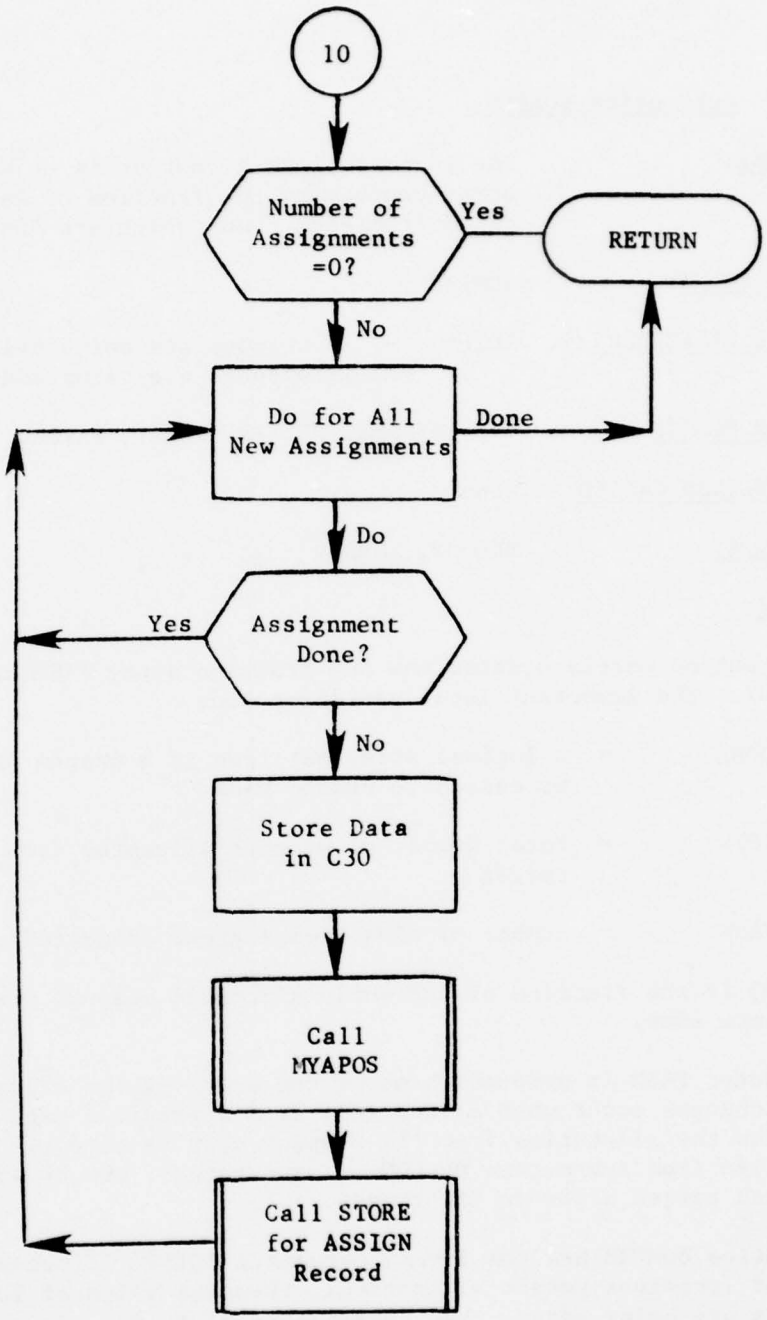


Figure 31. (Part 2 of 2)

3.8.3 Subroutine BOMPRM

PURPOSE: The purpose of this routine is to maintain the array containing the fraction of weapons all allocated from each group which are ASMs.

ENTRY POINT: BOMPRM

FORMAL PARAMETERS: IDIFF = -1 if weapons are being deleted
+1 if weapons are being added

COMMON BLOCKS: C33, DYNAMI, MULTIP, NALLY, PAYSAV, SALVO, WEPSAV

SUBROUTINES CALLED: None

CALLED BY: MULCON, SCNDGD

Method:

This routine merely updates the ASM fraction array FASM in common block /SALVO/. The important local variables are:

DONE = a logical array set true if a weapon has already been processed to update FASM

TOTW = total number of weapons allocated from a group on the target

TASM = number of ASMs from a group allocated on the target

FASM(G) is the fraction of currently allocated weapons from group G which are ASMs.

The factor FASM is updated whenever the state of the allocation changes. These changes occur when allocations from a previous pass are removed and when the allocation from the present pass is output. Thus, BOMPRM is called from subroutine MULCON on each target, and by subroutine SCNDGD for each target after the first pass.

Subroutine BOMPRM has one formal parameter IDIFF. If weapons are being removed (previous pass's allocation), then the value of IDIFF is -1. If weapons are being added, then IDIFF is equal to +1. In subroutine SCNDGD, the call to BOMPRM with IDIFF equal to -1 is made after reading the last pass allocation, just prior to the update of the running allocation sums. The call from MULCON with IDIFF equal to +1 is made just prior to the running sum update.

Upon entry to subroutine BOMPRM, the routine checks variable NBLN in /C33/. If this variable is negative, the allocation in /DYNAMI/ was made by subroutine DEFALOC and contains no bomber weapons. In this case, the subroutine returns with no further processing. If the

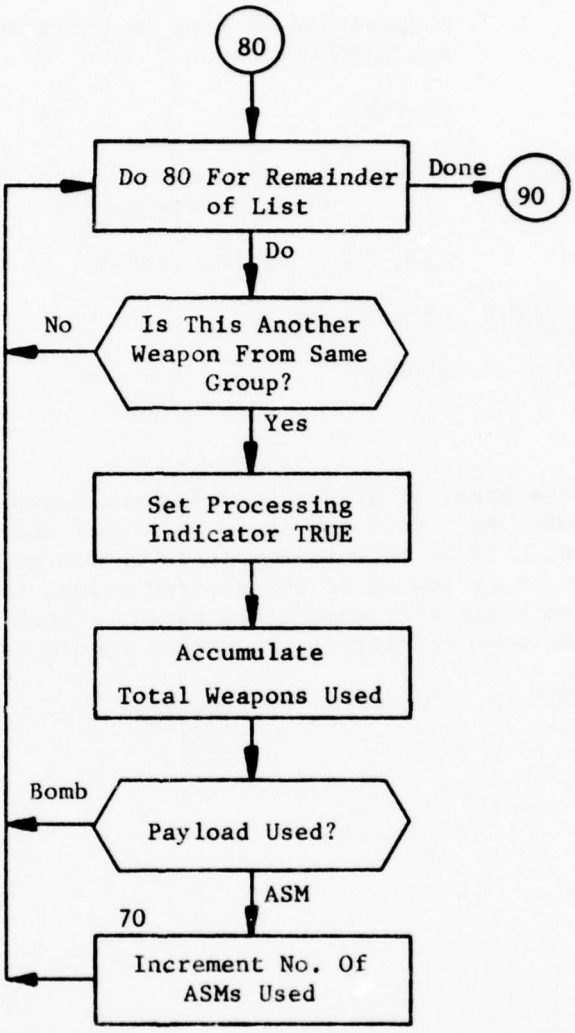


Figure 32. (Part 2 of 2)

3.8.4 Subroutine MYAPOS

PURPOSE: To position records properly before storage of a new ASSIGN record.

ENTRY POINTS: MYAPOS

FORMAL PARAMETERS: IMYAP: 1, Find weapon group and target
2, Find target only

COMMON BLOCKS: C10, C30, GRPHDR, TARREF

SUBROUTINES CALLED: DIRECT, NEXTTT

CALLED BY: ASGOUT

Method:

On first call the array of group IDS reference codes is set to zero. From then on, with each call the saved reference code of the desired group is checked. If it is nonzero it is retrieved. If it is zero, the group chain is cycled up to the desired group, the intervening groups also have their reference codes saved. Finally, when the proper group record has been retrieved, the target record is retrieved.

Subroutine MYAPOS is illustrated by figure 33.

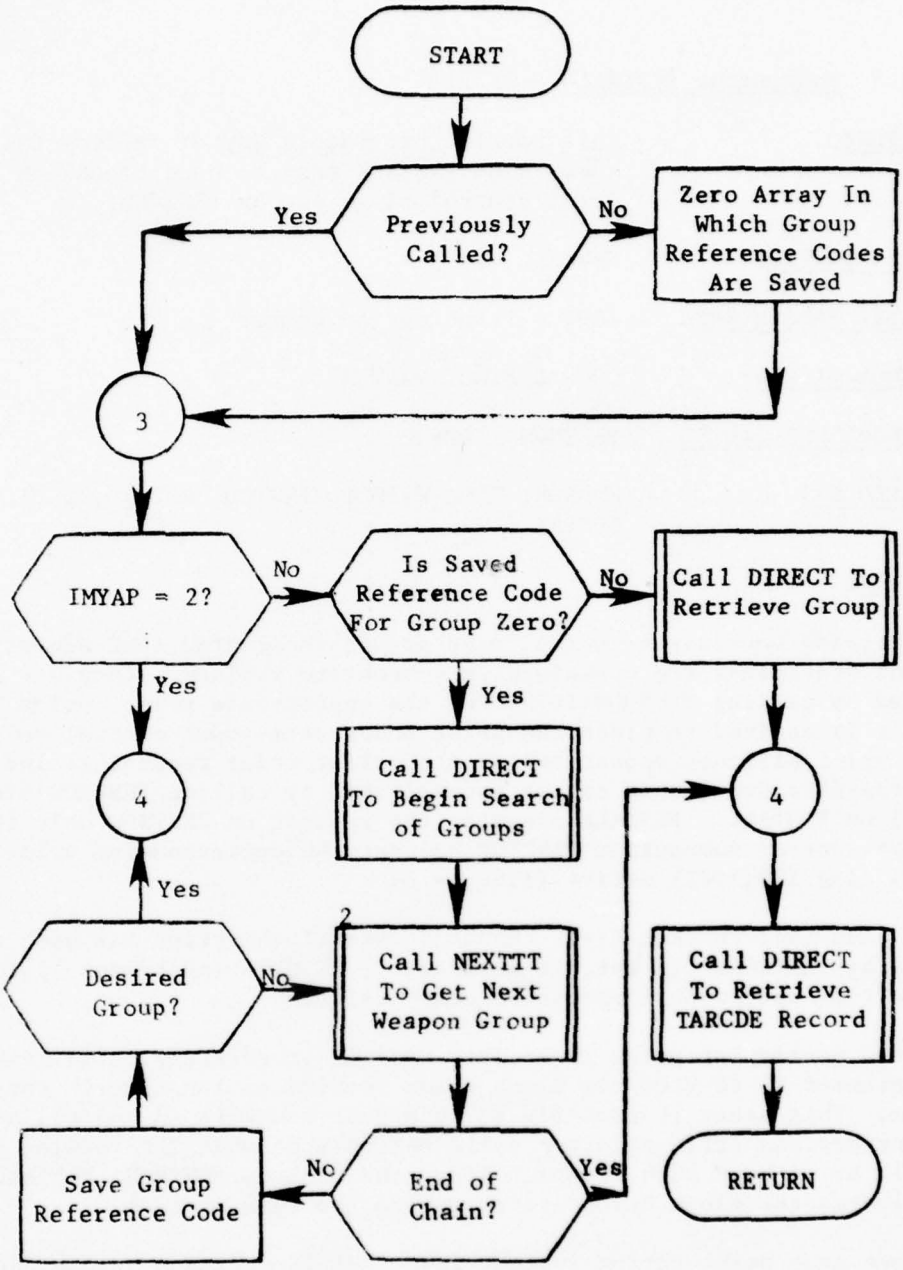


Figure 33. Subroutine MYAPOS

3.8.5 Subroutine PRNTALL

PURPOSE: This routine provides a way of calling the print subroutine PRNTNOW that is conditional on the print control flags set by PRNTCON.

ENTRY POINTS: PRNTALL

FORMAL PARAMETERS: IOPT - Print option number

COMMON BLOCKS: C30, CONTRO, PRNTCN

SUBROUTINES CALLED: PRNTNOW, TIMEME

CALLED BY: MULCON, WAD, WADOUT, FRSTGD, RESVAL, DEFALOC, SETPAY

Method:

To provide convenient control over prints in program ALOC almost all print statements are contained in subroutine PRNTNOW. They are activated by calling PRNTNOW(IOPT) for the appropriate print option IOPT. If it is desired to place the print under data-input control so that the print will not appear unless a specific print request is included in the data deck, this can be accomplished by calling PRNTNOW via a call on PRNTALL. PRNTALL executes the request on PRNTNOW only if the print control subroutine PRNTCON has set the corresponding print control flag IDO(IOPT) active (i.e., = 3).

For each call PRNTALL first checks to see if the print has been set active by PRNTCON. If not, it immediately RETURNs (statement 2) to minimize time wasted on inoperative print calls.

If the particular print is active, PRNTALL immediately calls TIMEME (statement 1) to stop the clock which records active time in the program. This makes it possible to do a test run with an unusual number of prints and still obtain a valid estimate of what the running time would be without such prints. After the call on PRNTNOW, PRNTALL reactivates the clock before returning to the main program.

Before each print option (except 26), PRNTALL prints a heading identifying the optional print.

Subroutine PRNTALL is illustrated in figure 34.

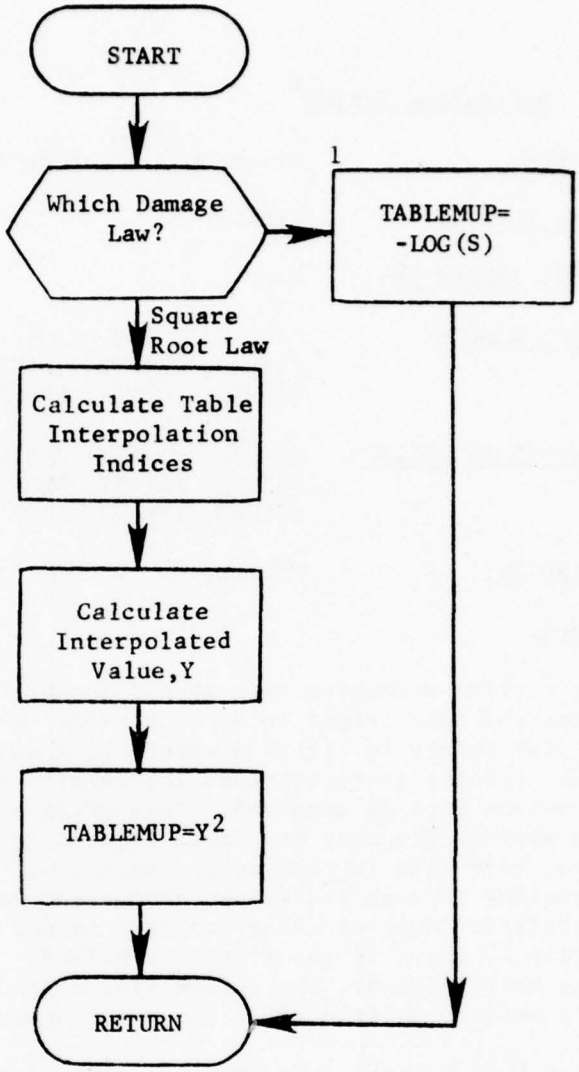


Figure 37. Function TABLEMUP

3.9 Subroutine FRSTGD*

PURPOSE: Assemble allocation data on the first pass.

ENTRY POINTS: FRSTGD

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C15, C30, C33, DYNAMI, FIL21, FILL, FIRST, GRPHDR, GRPSTF, INITSW, MULTIP, PAYSAV, PNAV, TARREF, TGTSAV, WADWPN, WEPSAV, WPFIX, XFPX

SUBROUTINES CALLED: CRDCAL, DIRECT, FLGCHK, HDFND, HEAD, INICRD, MODFY, NEXTTT, NXSPLT, PKCALC, PRNTAL, RECON, RETRV, TGTCRD, TIMEME

CALLED BY: MULCON

Method:

This routine processes each target in target number order. Each call causes the next target to be retrieved. Since each record on the target list points to either a target or a complex record, the next step in the process is to retrieve the remainder of the target data. Next the weapon data is acquired. This process depends to a great extent upon whether the user has saved file 15 from a previous run of ALOC. If so, this file is read in and unpacked. If not, the file is created by cycling through the weapon groups and calculating the various needed quantities. Much of the group data needed for this process is contained on file 23 where it was stored by DATGRP. If the user has specified range modifications, any information which differs from that on file 15 is written on file 22 in the same format.

During this process, the INACTIVE array is set. This array has an entry for each group and is either set to 0 or 100. 0 implies that the group is available for allocation to the target. 100 indicates that the group is unavailable for one of several reasons: target out of range, time decay requirements, and flag location and MIRV restriction. This array is written onto unit 21.

The final step is to read in any fixed assignments to the target and update the assignment records.

Subroutine FRSTGD is illustrated in figure 38.

---

* First subroutine of segment FGD.

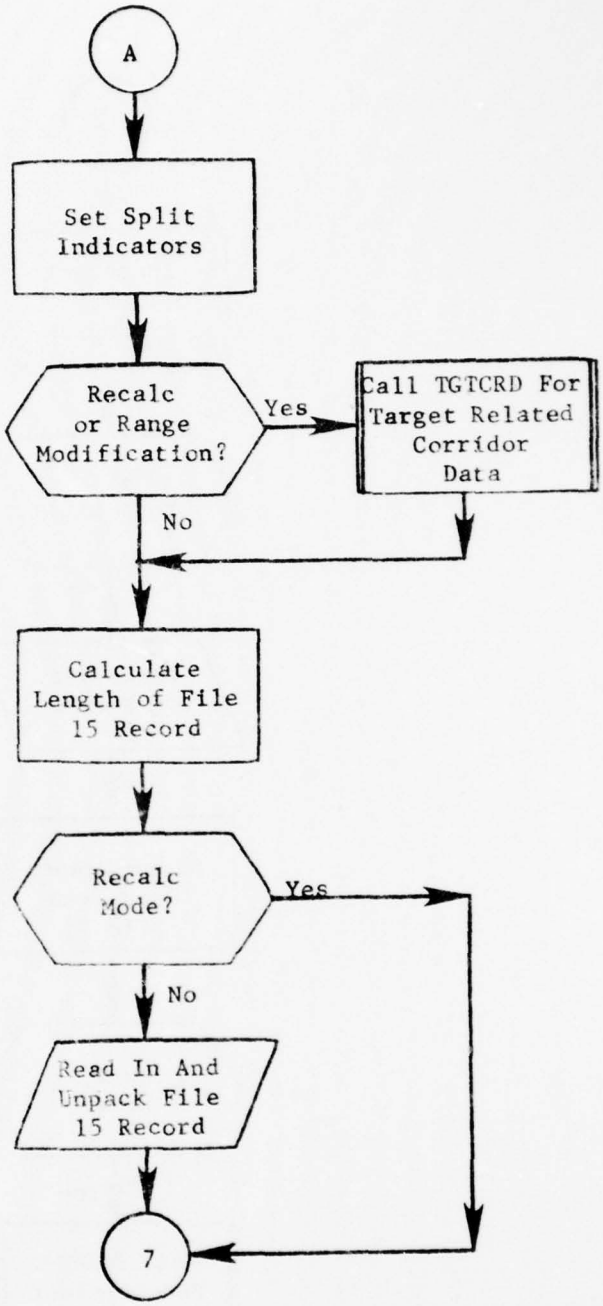


Figure 38. (Part 3 of 11)

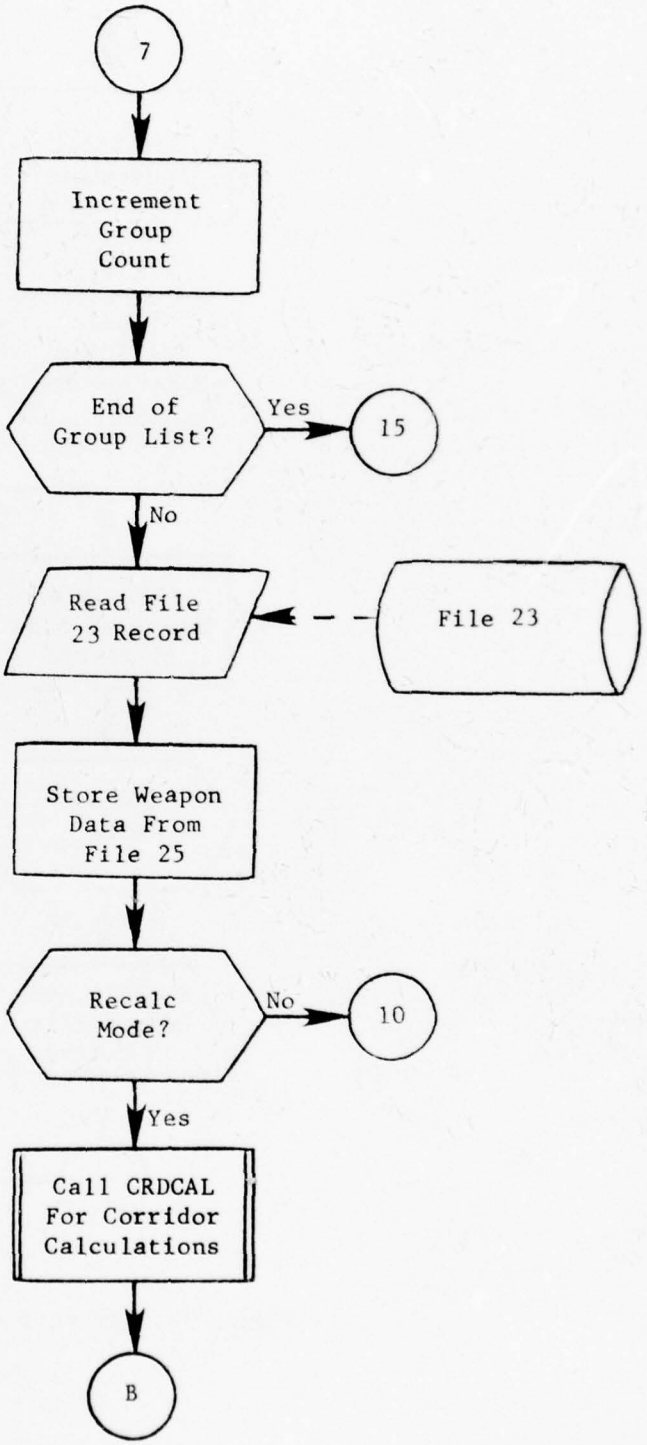


Figure 38. (Part 4 of 11)

SECTION 5. MODULE ALOCOUT

5.1 Purpose

Module ALOCOUT is responsible for selecting optimum DGZs (desired ground zeros), also called weapon aim points for weapon allocated to target complexes. ALOCOUT, also resorts weapon assignments at the group level for use within the Sortie Generation subsystem. If executed as part of mini-allocator process (DATAMAKE), produces a STRIKE format tape rather than resorting weapon assignments. (For format of STRIKE tape see CSM MM 9-77 Volume IV).

Module ALOC specifies weapon groups assigned to targets together with associated targeting data. ALOCOUT extracts data from these records and computes any aiming offsets required by the plan. For simple or multiple targets, no calculations are performed. In the case of complex targets which can have several elements at slightly different coordinates, ALOCOUT employs subroutine DGZ (desired ground zero selector) to select optimum aim points within the target complex.

5.2 Input

ALOCOUT operates after module ALOC assigned weapons to targets. These records (ASSIGN) in addition to the supporting data base structure must be defined for proper execution.

5.3 Output

No new data base records are created during the execution of ALOCOUT. However, the weapon assignment records (called ASSIGN) are modified in two ways. First, for assignments to complex targets or for assignments to cities with nonzero radius, offsets as determined within the module are included within the ASSIGN record. Second, the assignment records at the weapon group level are resorted for use within the Sortie Generation subsystem. For missile groups, the sort is based on decreasing values of attribute RVAL. For bomber groups, the order is based on penetration corridor index and within the corridor sorted based on attribute RVAL. The penetration corridor that contains the largest number of strikes appears first within the sort, followed by the penetration corridor of the next largest number of strikes and so on.

5.4 Concept of Operation

ALOCOUT (that is, subroutine ENTMOD) operates with three overlay links. The first overlay reads the target list (TARNUM) passes controls to subroutine PROCCOMP for offset calculations when applicable and finally supplies optional prints. After all targets have been processed controls passes to the second overlay which consists entirely of subroutine SUMPRN which reorders strikes at the weapon group level and, if requested, produces prints concerning the individual assignments. If part of mini-allocator, the second overlay is skipped and the third overlay is executed.

5.5 Identification of Subroutine Functions

5.5.1 Subroutine PROCCOMP. This subroutine controls the bulk of processing for offset determination. It is executed by subroutine ENTMOD only for those individual targets that require offset calculations. After offsets have been determined the assignment record (ASSIGN) is updated to include the values. Then, PROCCOMP returns to subroutine ENTMOD for acquisition of the next target and the associated strikes.

5.5.2 Subroutine SUMPRN. This subroutine constitutes the entire second overlay of ALOCOUT. Its purpose consists of resorting the weapon strikes at the group level and providing optional prints.

5.5.3 Subroutine MINIOUT. This subroutine controls the third overlay and produces a STRIKE tape.

5.6 Common Block Definition

Common blocks used by EVALALOC are outlined in table 11. Common blocks that communicate with the COP are given in appendix A of Program Maintenance Manual, Volume I.

Table 11. ALOCOUT Common Blocks
(Part 1 of 2)

BLOCK	VARIABLE OR ARRAY	DESCRIPTION
CITY	ICITY	Set to nonzero for targets with attribute RADIUS not equal to zero
C1	XO(J), YO(J)	Coordinates of target element J
	VI(J)	Initial target element values
	RADL(J)	Lethal radius of target element J
	VTOA(J,I)	Value of target element J immediately following arrival of weapon I
	S(J,I)	Survival probability of target element J relative to weapon I
	VEFF(J,I)	Effective value of target element J relative to weapon I
	X(I),Y(I)	Offset coordinates of weapon I
	PDEL(I)	Probability of delivery of weapon I
	ERDEL(I)	Error in delivery of weapon I
	YDSCL(I)	Scaled yield for weapon I
	VESC(I)	Intermediate computational value used in subroutine VAL for determination of total escaping target value
	NI	Number of weapons for complex
	NJ	Number of target elements for complex
GAMETIME	KDAY	Day set for STRIKE times (set to 1)
	KMON	Month set for STRIKE times (set to 1)
	KYEAR	Year set for STRIKE times (set to 1)
	HHR	Hour set for STRIKE times (set to 10)
GRPY	GROUPY(250)	Logical switch indicates if group allocated

Table 11. (Part 2 of 2)

BLOCK	VARIABLE OR ARRAY	DESCRIPTION
IONPRT	IPINDAT	User supplied print frequency for print option 1
	PRINCE(9)	Set TRUE if user requested option
ISKIPDGZ	ISKIPDGZ	Use indicator for DGZ. Normally it is 0. Compress resets it to 1 if more than 20 calls to it are made to reduce the number of target elements for a complex target; DGZ is not used again for the target in this case
JAZ	F(400,26)	Holding arrays for sort purposes
LOCFIN	LOCFIN	Starting location into IRSET's arrays for adverb FINDMIN instructions
STRIKE	TOA(I)	Weapon time of arrival to target
	IREFSTRK(I)	Reference code of weapon strike
	N	Number of strikes
WPGT	YDMIN	Minimum allowable weapon group yield
	IGRP	Group number

5.7 Subroutine ENTMOD

PURPOSE: Read user inputs, collect target weapon assignments, and execute subroutine PROCCOMP for DGZ determination.

ENTRY POINTS: ENTMOD (first subroutine executed when overlay ALOCOUT is called)

FORMAL PARAMETERS: None

COMMON BLOCKS: CITY, C10, C15, C30, GRPY, IONPRT, LOCFIN, OOPS, STRIKE, WPGT

SUBROUTINES CALLED: DGZ, DIRECT, HDFND, HEAD, INSGET, NEXTTT, PROCCOMP, RETRV, SUMPRN, TIMEME, WEPGET

CALLED BY: COP

Method:

Subroutine ENTMOD reads and stores users input, walks the individual target chain (TARNUM), collects weapon assignments for the current target, and calls subroutine PROCCOMP for DGZ determination if the target represents a complex or is a city (attribute RADIUS greater than zero). After processing all targets, subroutine SUMPRNT (second overlay) reorders weapon assignments on a weapon group basis for use within the Sortie Generation subsystem. Alternately, subroutine MINIOUT is called for miniallocator STRIKE tape.

Module ALOCOUT recognizes user supplied adverbs FINDMIN and ONPRINTS. FINDMIN sets the number of iterations subroutine FINDMIN uses for offset determination. ONPRINTS sets user options; results maintained in array PRINCE.

ALOCOUT now walks the identical target chain (TARNUM) which module ALOC made weapon assignments to. For each target, weapon assignments are stored on chain ASGWPN. If no strike exists processing continues by retrieving the next target on the list. Otherwise, for each weapon assignment, subroutine WEPGET retrieves weapon related attributes and updates necessary counts. Checks determine the nature of the target. Offsets are calculated only if the target represents a complex or is a city and has a nonzero RADIUS.

A complex target (or target complex) is a combination of target elements sufficiently close in geographic location that a weapon on any one of them will have some probability of killing other elements in the complex. Such target complexes are targeted as a unit by the allocator which allocates weapons against their total value, using one set of coordinates. In order to maximize targeting efficiency against such a complex, optimum aim points among the target elements must be selected. These aiming offsets are specified relative to the first target element only and are passed on in that form to subsequent modules.

When ENTMOD encounters a complex target, subroutine PROCCOMP is called. PROCCOMP is responsible for assembling the data on a complex target in a form that can be used efficiently for DGZ selection. Each target component of the complex generates a standardized target element in the arrays used by DGZSEL. (Targets with more than one hardness component generate more than one such target element, and targets with a specified target radius will generate several elements spread over the area of the target to represent a value spread over the area.)

Subroutine ENTMOD is illustrated in figure 68.

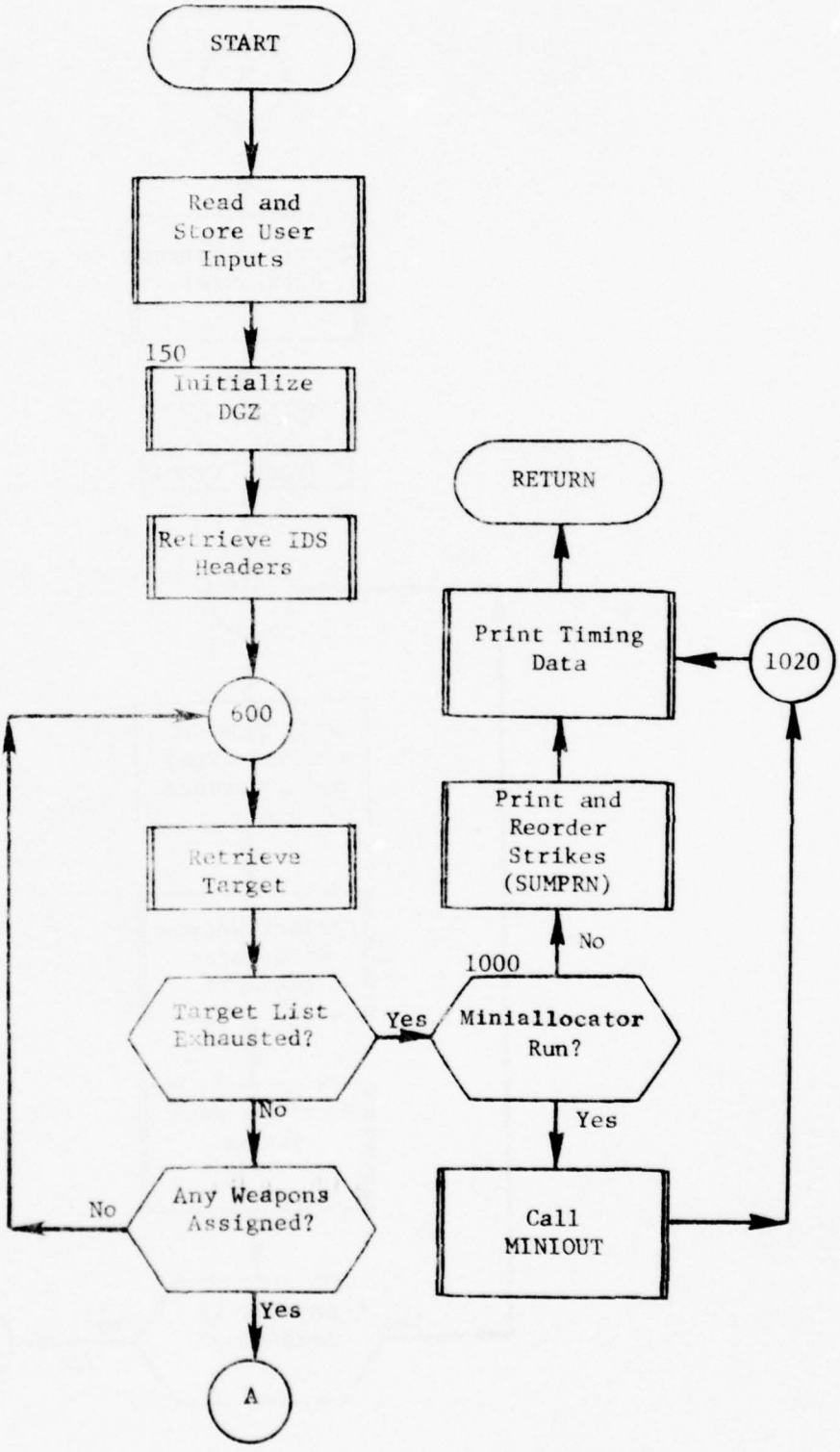


Figure 68. Subroutine ENTMOD (Part 1 of 3)

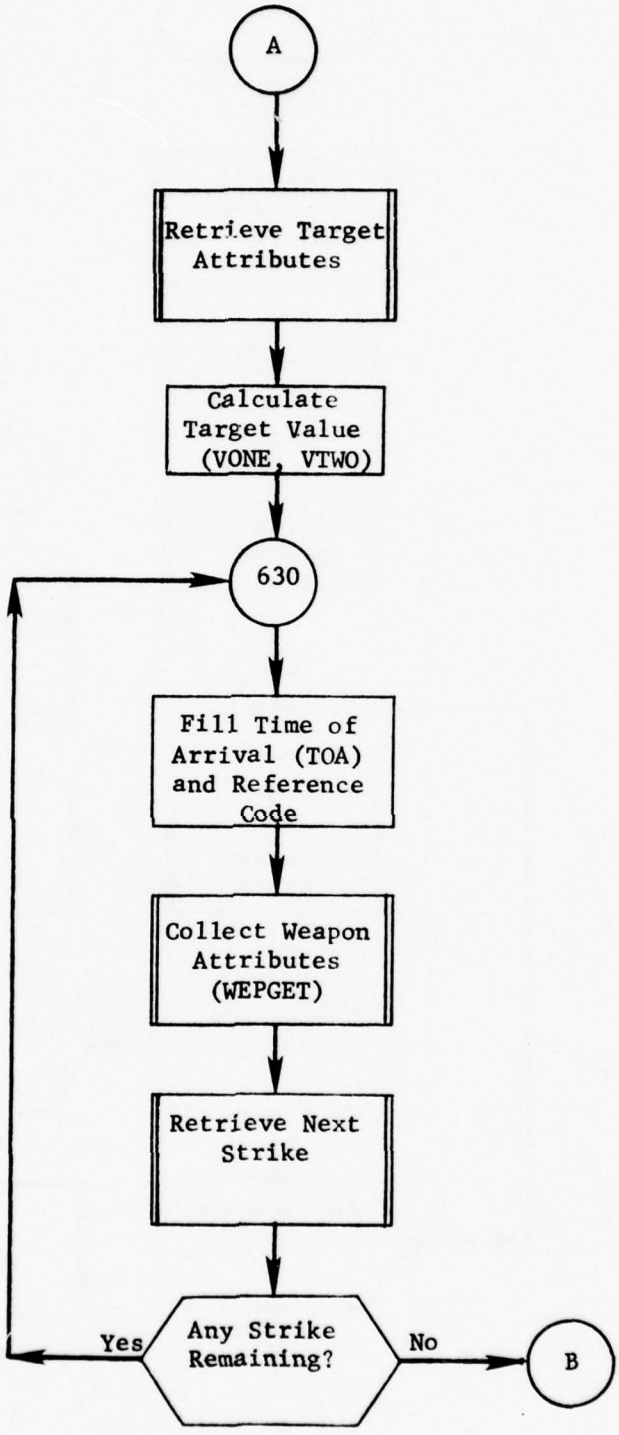


Figure 68. (Part 2 of 3)

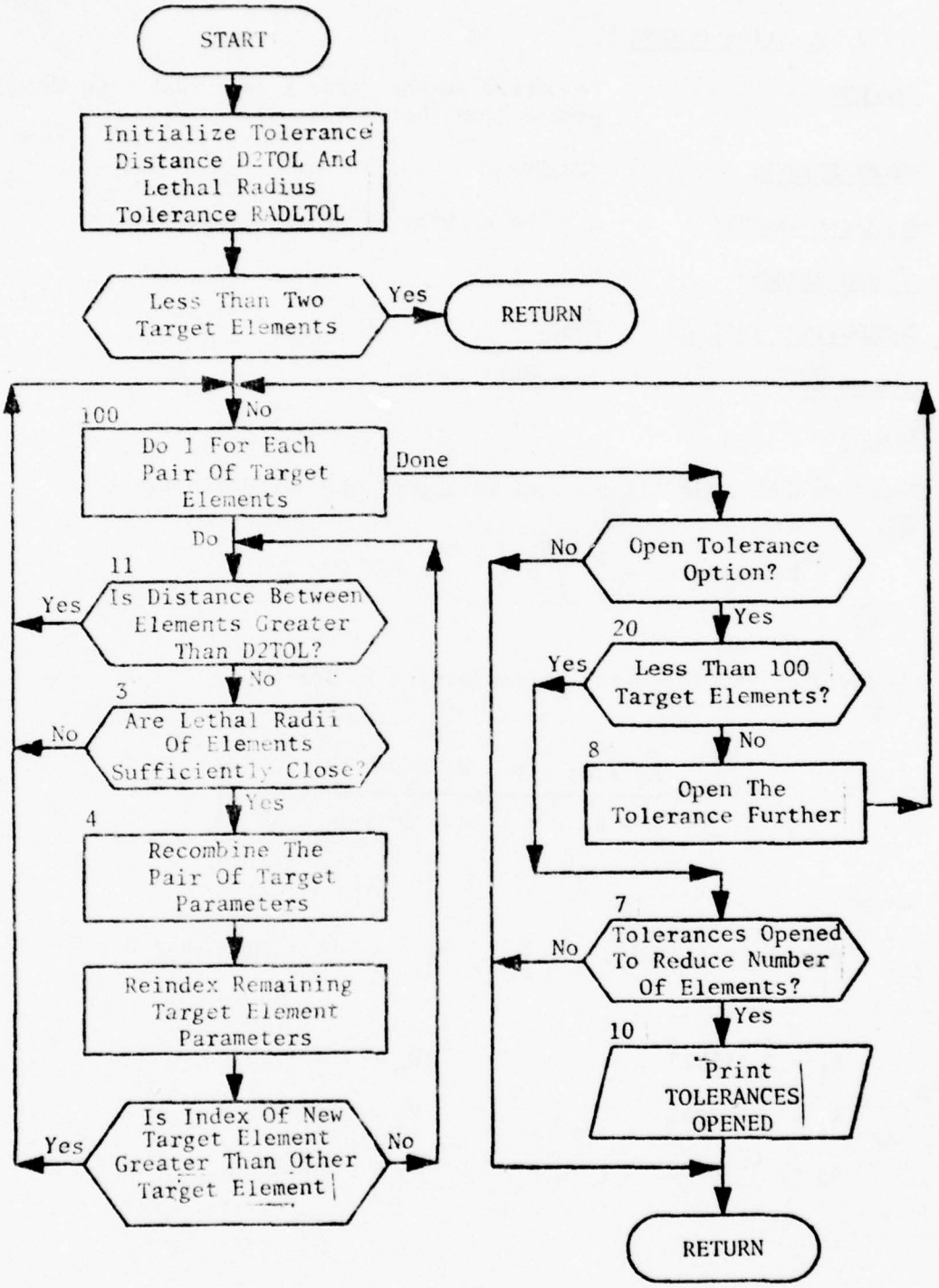


Figure 69. Subroutine COMPRESS

5.7.2 Function CUMINV

PURPOSE: To determine the value X such that Z is the probability that $x \leq X$.

ENTRY POINTS: CUMINV

FORMAL PARAMETERS: Z - The probability that $x \leq X$

COMMON BLOCKS: None

SUBROUTINES CALLED: None

CALLED BY: PROCCOMP

Method:

Function CUMINV is illustrated in figure 70. By definition,

$$Z = P[x \leq X] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{X} e^{-\frac{t^2}{2}} \cdot dt \text{ for } 0 < Z < 1$$

CUMINV uses the following approximation X^1 for X:

$$X^1 = \pm \left[V - \left(\frac{A_1 + A_2 \cdot V + A_3 \cdot V^2}{1 + B_1 \cdot V + B_2 \cdot V^2 + B_3 \cdot V^3} \right) \right]$$

where

$$V = \sqrt{\ln (1/Q^2)}, \quad Q = Z \text{ or } 1 - Z \text{ such that } 0 \leq Q \leq .5$$

and

$A_1 = 2.515517$ $B_1 = 1.432788$

$A_2 = .802853$ $B_2 = .189269$

$A_3 = .010328$ $B_3 = .001308$

5.7.10 Subroutine PROCCOMP

PURPOSE: To set up arrays in common block /C1/ for the complex target so that the subroutine DGZ can use the arrays during the selection of optimal aim point offsets for the weapons allocated to the target; and to modify target weapon assignments records for inclusion of the computed offsets.

ENTRY POINTS: PROCCOMP

FORMAL PARAMETERS: None

COMMON BLOCKS: CITY, C1, C10, C30, ISKIPD, STRIKE, WPGT

SUBROUTINES CALLED: COMPRESS, CUMINV, DGZ, DIRECT, ERGOT1, ERGOT2, HEAD, MODFY, NEXTTT, ORDER, REORDER, TIMEME, VALTAR

CALLED BY: ENTMOD

Method:

When ENTMOD encounters a complex target, PROCCOMP is called in order to assemble data in a form that can be efficiently used for DGZ selection. Each target component of the complex generates a standardized "target element" in the working arrays used by subroutine DGZ (common /C1/). Targets with more than one hardness component generate more than one such target element, and targets with a specified target radius will generate several elements spread over the area of the target to represent a value over the area. For complexes, individual target elements are obtained by walking the data base chain called 'CMPTGT'.

If the number of target elements so generated exceeds the maximum program dimensions (120), subroutine COMPRESS is called to recombine target elements near each other having nearly the same lethal radius. In any case, for efficiency in DGZ, a call to COMPRESS is made just before calling DGZ. On return from DGZ, PROCCOMP modifies weapon assignment records (ASSIGN) for definition of the computed offsets.

Subroutine PROCCOMP is illustrated in figure 79.

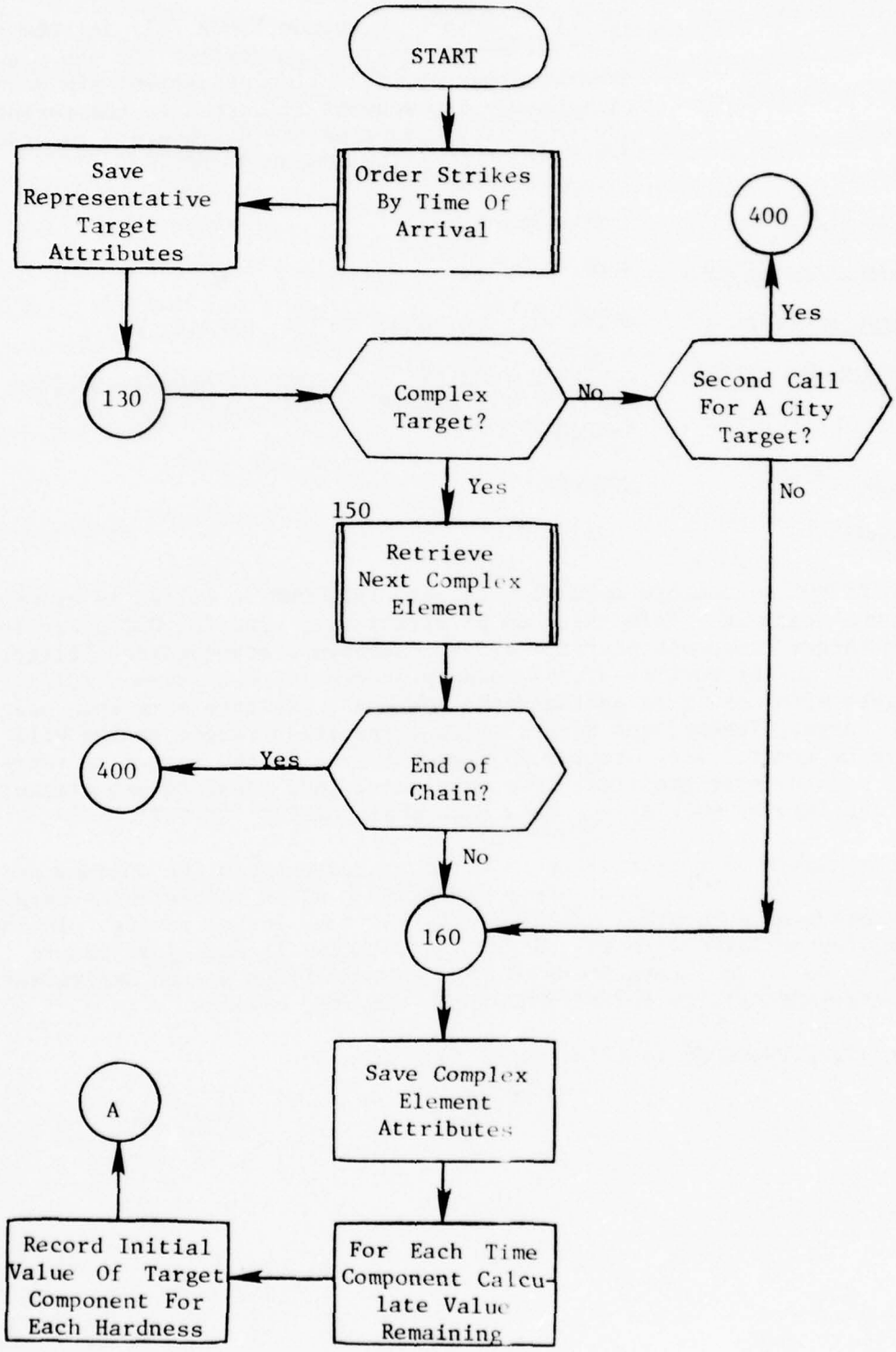
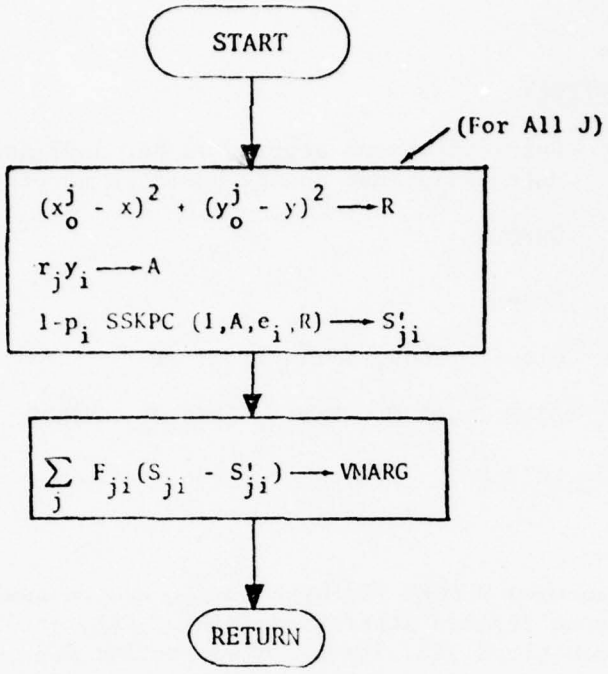


Figure 79. Subroutine PROCCOMP (Part 1 of 4)



(x,y) = Position of Weapon I

(x_o^j, y_o^j) = Coordinates of Target Element J

r_j = Lethal Radius, Target J

y_i = Scaled Yield, Weapon I

p_i = Probability of Delivery, Weapon I

e_i = Error in Delivery, Weapon I

S_{ji} = Survival Probability of Target J Relative to Weapon I

S'_{ji} = Survival Probability of Target J Relative to Weapon I when it is Assigned to Position (x,y)

Figure 82. Function VMARG

5.7.14 Subroutine WEPGET

PURPOSE: Retrieve weapon attributes per assignment, and update assignment counts based on corridor.

ENTRY POINTS: WEPGET

FORMAL PARAMETERS: None

COMMON BLOCKS: C1, C10, C30, GRPY, WPGT

SUBROUTINES CALLED: DIRECT, HEAD, NEXTTT, RANSIZE, TIMEME

CALLED BY: ENTMOD

Method.

Subroutine ENTMOD executes WEPGET for each target weapon assignment. WEPGET retrieves weapon related attributes (YIELD, CEP, etc.) and defines arrays in common block /C1/ for use by subroutine DGZ. Also, a count of each weapon assignment categorized by group and penetration corridor is updated.

The weapon attributes necessary for calculating offsets for each strike are indexed at the weapon group level. That is, individual strikes launched from the same weapon group have the same attribute values. Therefore it is necessary to interface with the data base only once per weapon group request. Upon initial extraction, these attributes are written on an indexed random file for future reference.

Subroutine WEPGET is illustrated in figure 83.

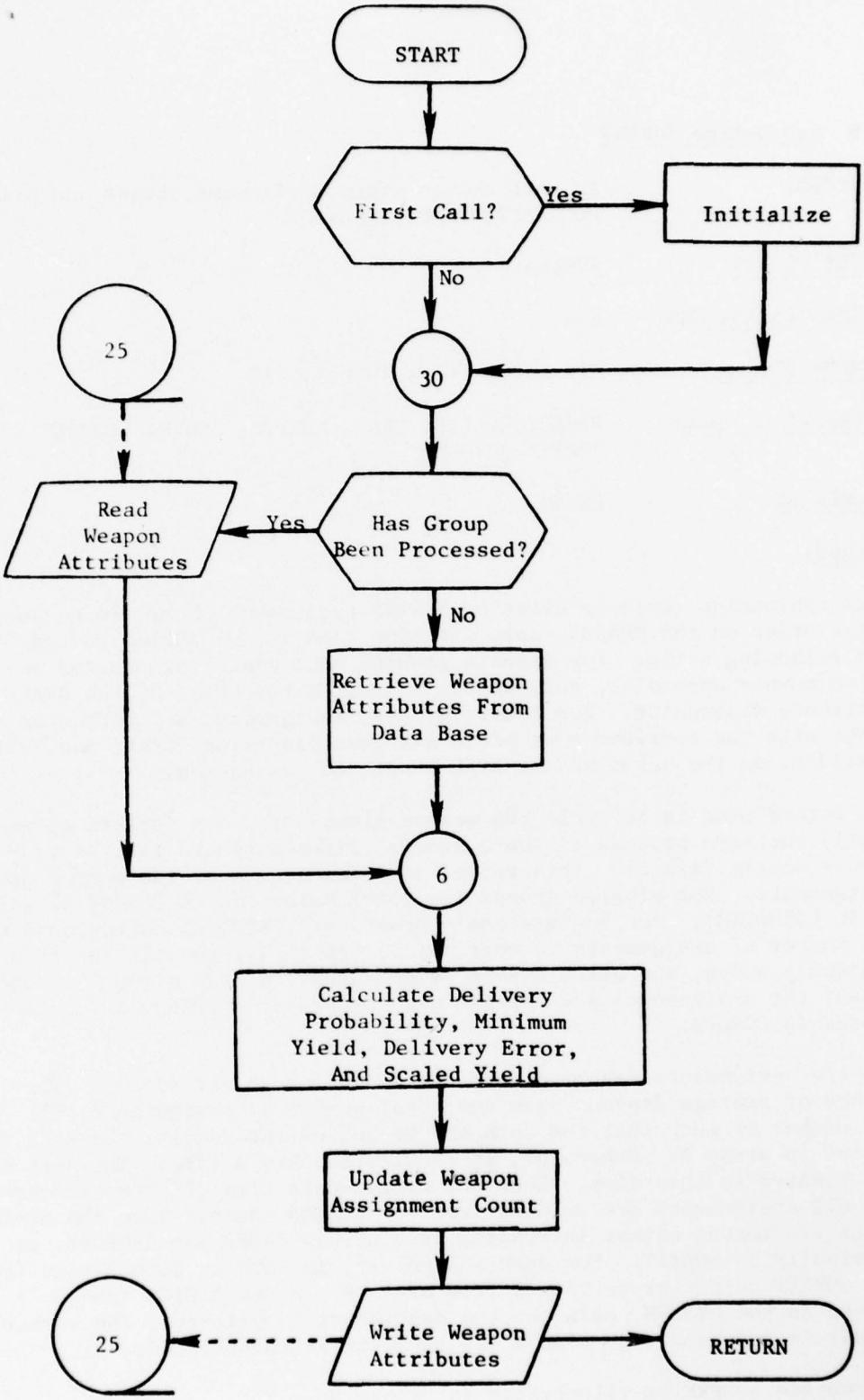


Figure 83. Subroutine WEPGET

5.8 Subroutine SUMPRN

PURPOSE: To sort weapon group assignment chains and print out assignment summaries

ENTRY POINTS: SUMPRN

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C30, GRPY, IONPRT, JAZ

SUBROUTINES CALLED: DIRECT, DLETE, HEAD, NEXTTT, ORDER, REORDER, SORTIT, TIMEME

CALLED BY: ENTMOD

Method:

This subroutine replaces existing ASSIGN records which are in no particular order on the MYASGN chain with the same set of ASSIGN records in the following sorts: For missile groups, assignments are sorted on salvo number ascending, and, within salvo, on the value of the RVAL attribute descending. For bomber groups, assignments are sorted on corridor with the corridor most often assigned occurring first, and, within corridor, on the value of the RVAL attribute descending.

The method used is to cycle the weapon group chain and perform essentially the same process for each group. First a record is read from random access file 25. This record contains counts of the weapon groups assignments. For missile groups the total number is in CORCNT(1) and -1 in CORCNT(2). For bombers the contents of CORCNT(I) corresponds to the number of assignments to corridor I. At this point, if the group is a missile group, the print header is produced. If the group is a bomber group, the assignments are totaled and the proper corridor order is stored in CORORD.

Now the assignments are read from the MYASGN chain and stored. The method of storage depends upon the total number of assignments. If the number is such that the data may be sorted internally, the data is stored in array F. Otherwise, it is written onto a file. The sort keys are created at this time. When the assignments have all been processed, the old assignments are deleted from the MYASGN chain. Then the assignments are sorted either internally by routines ORDER and REORDER, or externally by SORTIT. Now each assignment, in sort is either read from the SORTIT output or retrieved from array F. A new ASSIGN record is stored in the MYASGN chain and the assignment is printed. The bomber header is produced each time a new corridor is encountered.

Subroutine SUMPRN is illustrated in figure 84.

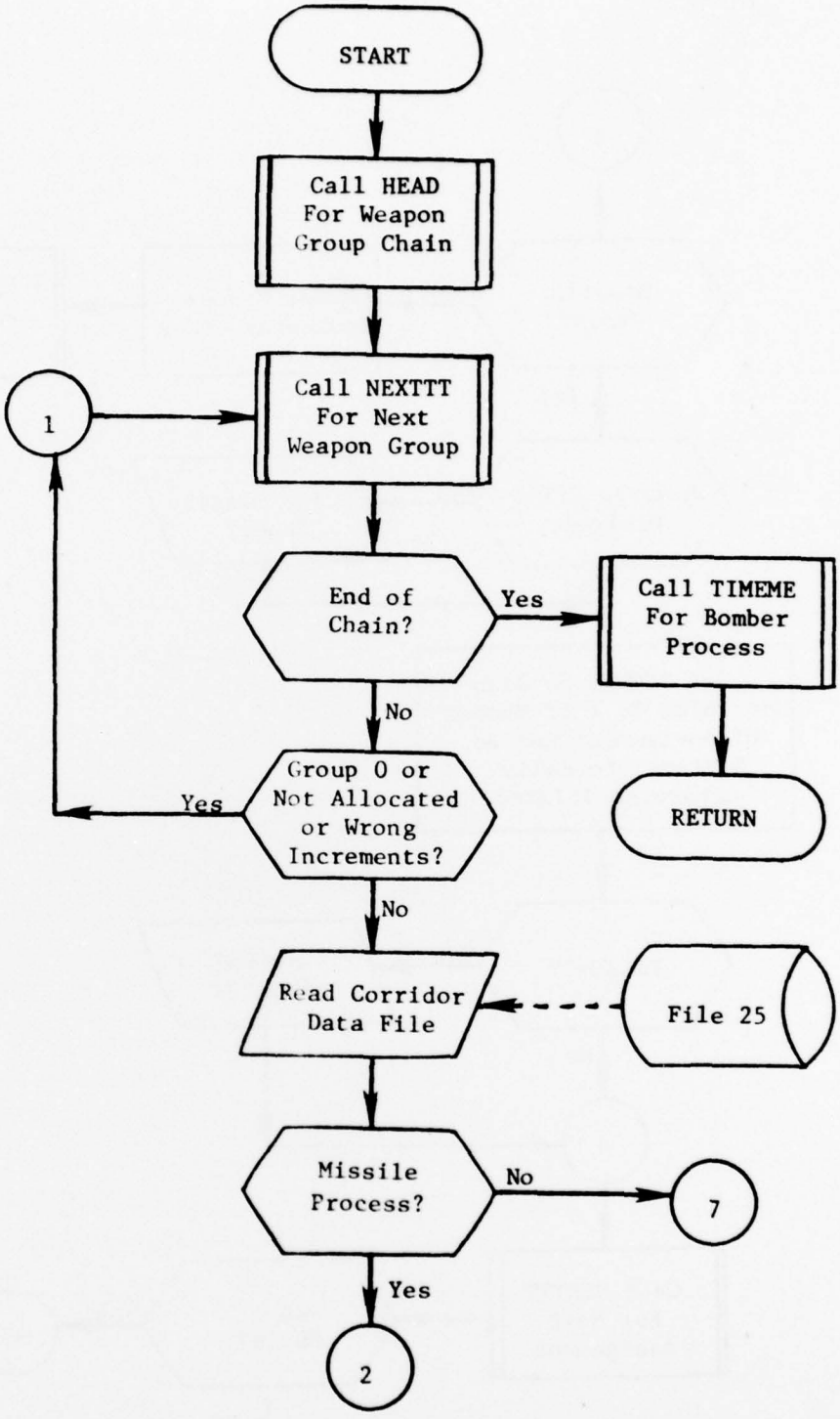


Figure 84. Subroutine SUMPRN (Part 1 of 9)

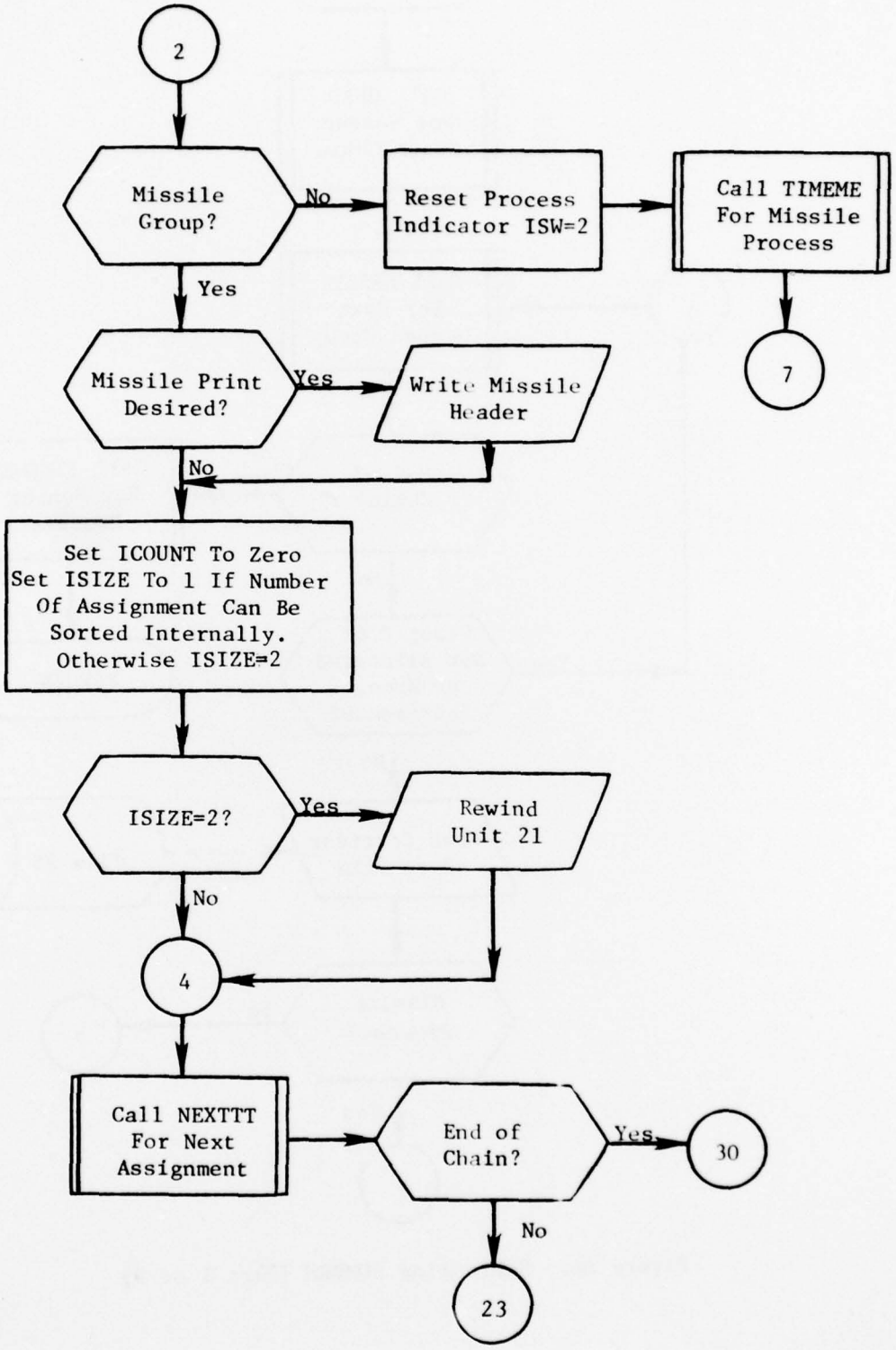


Figure 84. (Part 2 of 9)

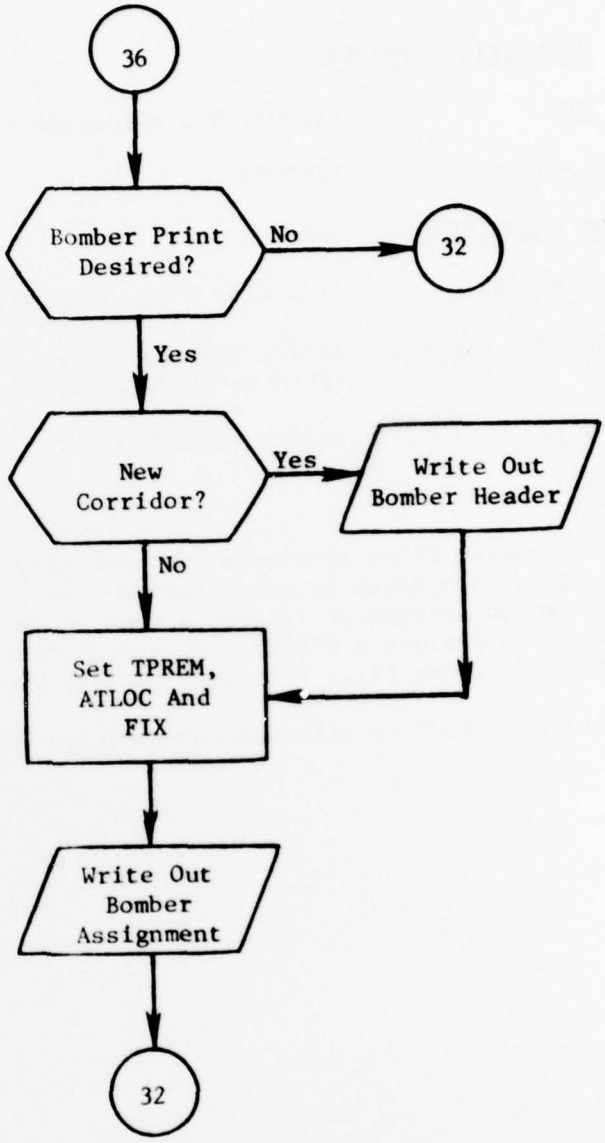


Figure 84. (Part 9 of 9)

5.9 Subroutine MINIOUT*

PURPOSE: Produce miniallocator STRIKE tape

ENTRY POINTS: MINIOUT

FORMAL PARAMETERS: None

COMMON BLOCKS: C10, C15, C30, GAMETIME, IONPRT

SUBROUTINES CALLED: ABORT, CONVLL, DISTF, FINDTIME, HDFND, HEAD, IGETHOB, INFORM, IPROB, NEXTTT, RETRV

CALLED BY: ENTMOD (ALOCOUT)

Method:

This subroutine first retrieves the necessary weapon, payload and warhead data. Then each group is processed in order. For each group, every associated weapon assignment record is used to develop and reformat the necessary data to produce a STRIKE tape record (for STRIKE tape format consult CSM MM 9-77 Volume IV). This tape is thus produced.

Subroutine MINIOUT is illustrated in figure 84.1.

* Main subroutine of overlay MINIO.

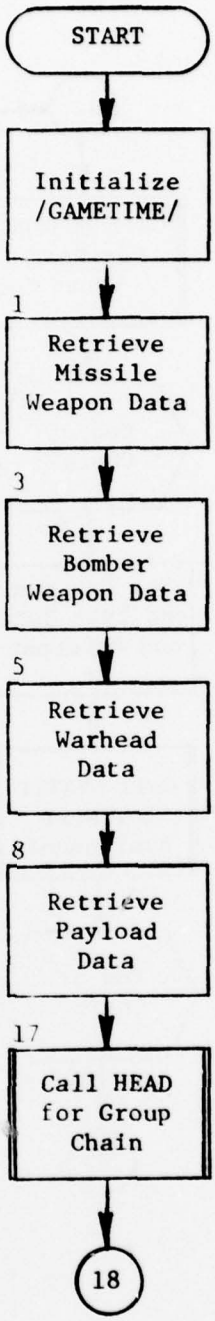


Figure 84.1. Subroutine MINIOUT (Part 1 of 3)

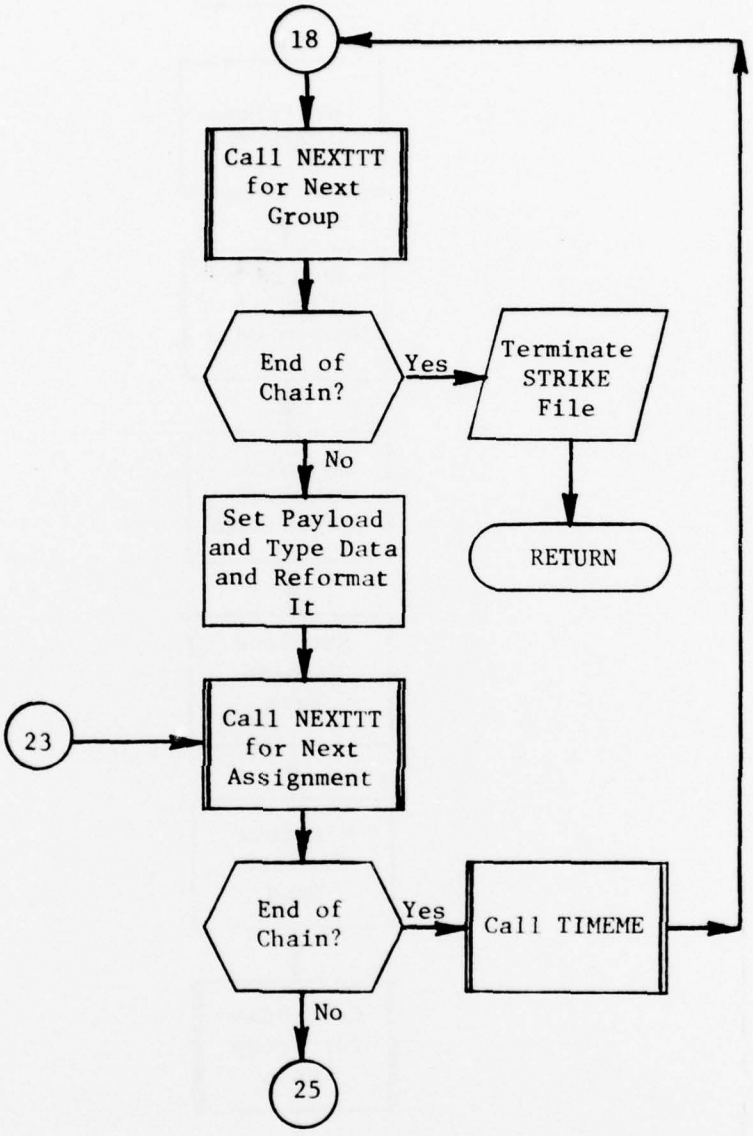


Figure 84.1. (Part 2 of 3)

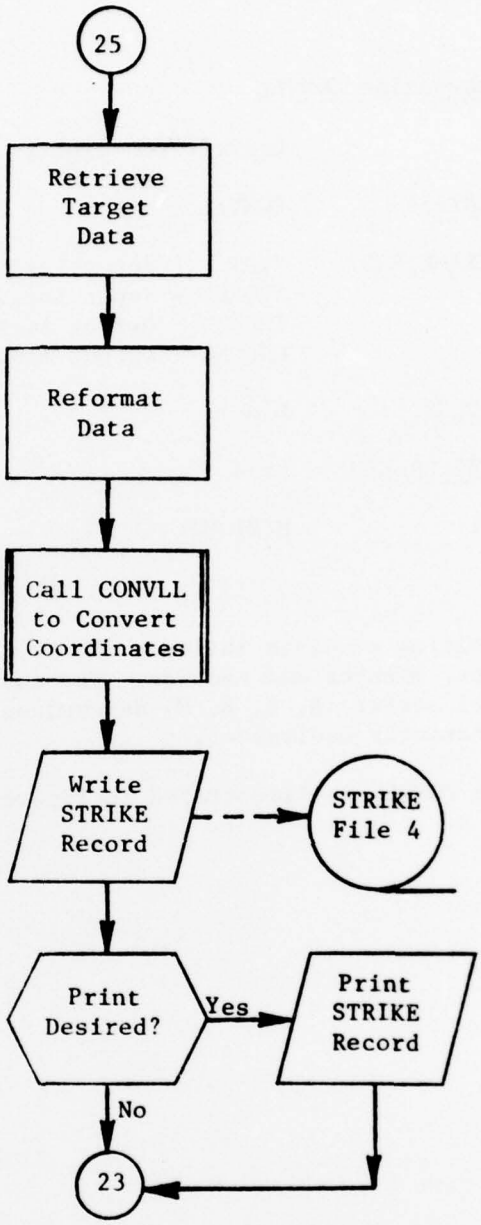


Figure 84.1. (Part 3 of 3)

5.9.1 Subroutine CONVLL

PURPOSE: Convert coordinates to character format

ENTRY POINTS: CONVLL

FORMAL PARAMETERS:
XLAT - Input latitude (degrees)
XLONG - Input longitude (degrees)
CHLAT - Output latitude (DDMMSSX)
CHLONG - Output longitude (DDDMMSSX)

COMMON BLOCKS: None

SUBROUTINES CALLED: None

CALLED BY: MINIOUT

Method:

This subroutine converts latitude then longitude. For each, it breaks out degrees, minutes and seconds. The quantities are checked and the directional letter (N, S, E, W) determined. Then ENCODE is used to create the character equivalent.

Subroutine CONVLL is illustrated in figure 84.2.

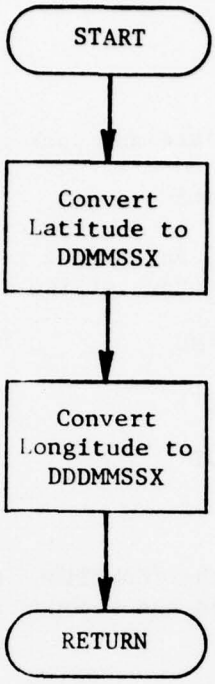


Figure 84.2. Subroutine CONVLL

5.9.2 Subroutine FINDTIME

PURPOSE: Calculate and pack strike time

ENTRY POINTS: FINDTIME

FORMAL PARAMETERS: XX - input strike time
II - output strike time (packed)

COMMON BLOCKS: GAMETIME

SUBROUTINES CALLED: None

CALLED BY: MINIOUT

Method:

Day and month are set from the /GAMETIME/ block. HHR is added to the input time and hours, minutes and seconds calculated. The results are packed into II and returned.

Subroutine FINDTIME is illustrated in figure 84.3.

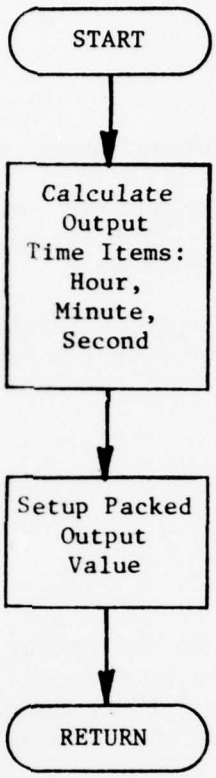


Figure 84.3. Subroutine FINDTIME

5.9.3 Subroutine INFORM

PURPOSE: To reformat numeric data

ENTRY POINTS:
INFORM - Numeric data
IPROB - Probabilities
NTIME - Times

FORMAL PARAMETERS:
IN,X,T - Input values
OUT - Output BCD value

COMMON BLOCKS: None

SUBROUTINES CALLED: None

CALLED BY: MINIOUT

Method:

All entry points eventually use the INFORM code which converts the absolute value of the numeric data to BCD and, if negative, adds a sign. The IPROB entry point returns "-1" if the input exceeds 100 percent. The NTIME entry point packs up the input time value in hours, minutes, and seconds.

Subroutine INFORM is illustrated in figure 84.4.

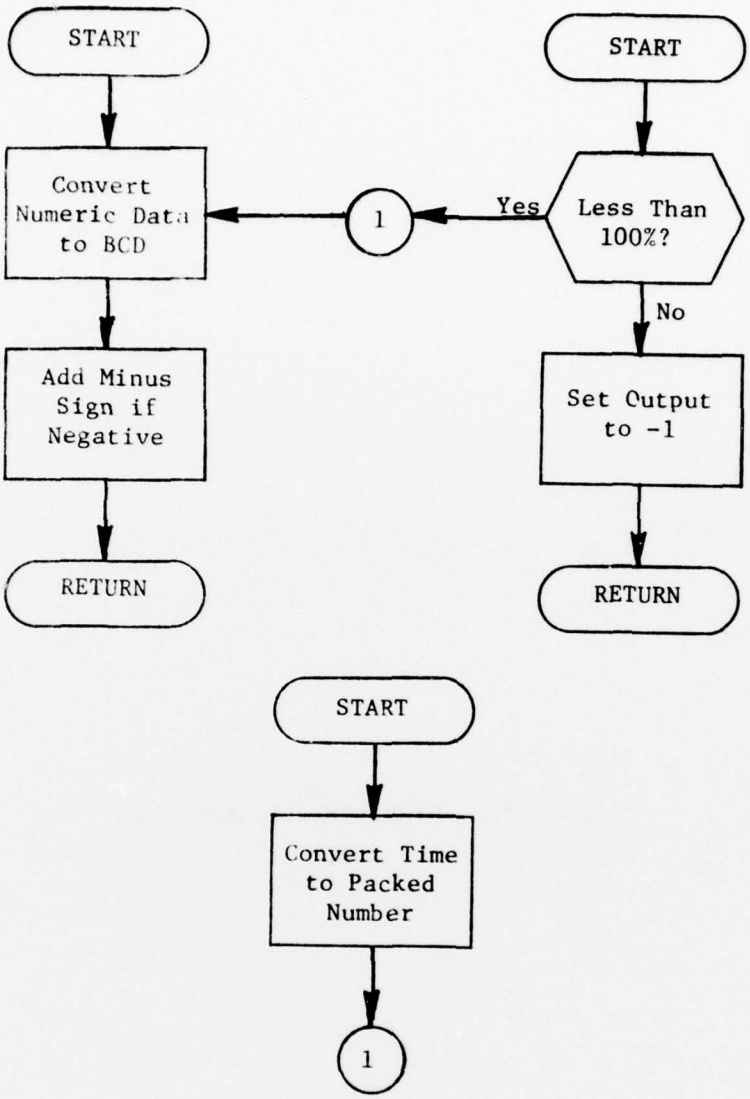


Figure 84.4. Subroutine INFORM